CU-CSSC-93-16

CENTER FOR SPACE STRUCTURES AND CONTROLS

Final Report to NASA/JSC (NAG 9-574)
REAL-TIME DYNAMICS AND CONTROL STRATEGIES FOR SPACE OPERATIONS OF FLEXIBLE STRUCTURES
(Period: January 15, 1992 - July 14, 1993)

by

K. C. PARK, K. F. ALVIN and S. ALEXANDER

JULY 1993

COLLEGE OF ENGINEERING
UNIVERSITY OF COLORADO
CAMPUS BOX 429
BOULDER, COLORADO 80309

SUMMARY

This project (NAG 9-574) was meant to be a three-year research project. However, due to NASA's reorganizations during 1992, the project was funded only for one year. Accordingly, every effort was made to make the present final report as if the project was meant to be for one-year duration. Originally, during the first year we were planning to accomplish the following: (1) we were to start with a three-dimensional flexible manipulator beams with articulated joints and with a linear control-based controllers applied at the joints; (2) using this simple example, we were to design the software systems requirements for real-time processing, introduce the streamlining of various computational algorithms, perform the necessary reorganization of the partitioned simulation procedures, and assess the potential speed-up realization of the solution process by parallel computations.

The three reports included as part of the final report address: (1) the streamlining of various computational algorithms; (2) the necessary reorganization of the partitioned simulation procedures, in particular the observer models; and (3) an initial attempt of reconfiguring the flexible space structures. We wish to state that much of the real-time effort via parallel computations will continue under a NSF grant as part of *High Performance Parallel Computing Initiative.* Other two aspects also constitute important attributes for real-time simulation of space operations. As such, they will also be pursued through other grants in the near future.

We thank Dr. John Sunkel for encouraging us through the difficult period of what turned out to be a short, intense yet productive endeavor.

9

Recent Developments in Time Integration Methods for Structural and Interaction System Dynamics

K. C. Park

University of Colorado at Boulder, USA

9.1 INTRODUCTION

This survey is a follow-up on earlier ones (Park [1]; Park and Felippa [2]; Felippa and Park [3]; Park [4]) on direct time integration methods. The algorithmic characterization of the integration formulae offered therein, namely, stability, accuracy and implementation aspects, remains largely intact. Readers wishing to familiarize the algorithmic characterization may refer to the references cited above plus Hughes and Belytschko [5].

What we are about to cover herein reflects a steady shift of research thrusts in computational dynamics since the mid-1980s, from discipline-oriented dynamics to system-oriented dynamics, from sequential computations to parallel computations, and from efficiency/accuracy concern to system model improvements/refinements. The specific topics we survey in this chapter thus reflect their maturing stages; these developments do not fit into a coherent theory or categorization at the present time.

Since time integration algorithms have been presented within the context of linear structural dynamics for most instances, first we report on computational methods for non-linear multibody dynamics. Major challenges in the development of computational

Solving Large-scale Problems in Mechanics, Edited by M. Papadrakakis

methods for multibody dynamics analysis have been the conservation of both energy and momentum, system constraint violations, and simulation speed. We will address some of these issues.

The second topic we will report is methods for the solution of coupled-field problems, primarily methods for control-structure interaction (CSI) problems. The design, modelling, analysis and real-time operation of CSI systems are one of the most intensely researched activities in recent years with applications ranging from aeroelastic tailoring, vibration control of reflectors deployed in low-earth orbits, to active vibration control of suspension systems. The third topic we will present is a computational method for transient thermal–structure interaction problems. This technique is relevant to the analysis of the thermal response of high-speed transport plane as well as integrated electronic chip thermal management problem.

9.2 SOLUTION TECHNIQUES FOR MULTIBODY DYNAMICS

The equations of motion for multibodies are characterized by two key features: highly nonlinear kinematical relations and complex constraints. It is *not* the purpose of this chapter to make an exhaustive survey of available solution techniques. Rather, we will examine selected techniques that meet our needs: computer implementability, adaptation to large-scale simulation, robustness and efficiency, in that order.

There are three aspects of solution techniques for multibody dynamics (MBD) analysis. First, we must have at hand an efficient and accurate algorithm for updating the kinematical quantities such as angular orientations, angular velocities. Second, direct time integration of the equations of motion that correspond to the unconstrained states of multibodies must be performed. Third, an accurate and efficient treatment of constraints is essential if the numerical solution is to maintain the given holonomic and nonholonomic constraints. In practice, the three aspects are intertwined so that one must achieve a careful balance in the employment of three computational aspects. As computer implementation of the three require different strategies, we will discuss them separately first and bring them together in the solution procedures.

The numerical solution procedure for MBD systems which we describe herein is termed a *staggered MBD solution procedure* that solves the generalized coordinates in a separate module from that for the constraint force (Park and Chiou [6]; Park *et al.* [7, 8]). A major advantage of such a partitioned solution procedure is that additional analysis capabilities such as active controller and design optimization modules can be easily interfaced without embedding them into a monolithic program. The solution of the equations of motion for constrained multibody systems, unlike typical structural dynamics problems, must satisfy at each time step the system constraints, whether holonomic or non-holonomic or time-specified manoeuvres. Because of this distinctive requirement, the reliability and cost of a multibody analysis package can be strongly affected by how efficiently and accurately the constraints are preserved during the numerical solution stage.

The system constraint forces can be either eliminated or retained depending upon the complexities associated with the elimination process. In general, it is preferable to eliminate the constraint degrees of freedom if they are associated only with open rigid links. On the other hand, if external torque or active control devices are attached to those joints, it is computationally more advantageous to solve the constraint forces (or Lagrange multipliers) simultaneously together with the generalized coordinates. Unfortunately, a straightforward way of computing the Lagrange multipliers can often lead to an unacceptable level of errors. The task of minimizing the propagation error due to violations of the constraint conditions is known as *stabilization*. We will describe a particular constraint stabilization which recasts the algebraic constraint conditions to a set of parabolic differential equations such that the constraint forces can also be integrated in time.

To solve for the generalized coordinates of the multibody system, the equations of motion are partitioned according to the translational and the rotational coordinates. This sets the stage for an efficient treatment of the rotational motions vïa the singularity-free Euler parameters. The translational part of the equations of motion is integrated via a standard central difference algorithm. The rotational part is treated by a modified central difference algorithm in order to preserve the discrete angular momentum. Once the angular velocities are obtained, the angular orientations are updated via the mid-point implicit formula employing the Euler parameters.

When the two algorithms, namely, the modified central difference algorithm for the rotational coordinates and the implicit staggered procedure for the constraint Lagrange multipliers, are brought together in a staggered manner, they constitute a staggered explicit-implicit procedure as detailed below.

9.2.1 *Equations of motion for multibody systems*

To motivate ourselves for the development of solution procedures for the multibody dynamics problems, let us introduce the following equations of motion:

$$\mathbf{M}\ddot{\mathbf{d}} = \mathbf{Q} - \mathbf{B}^T\lambda, \qquad \dot{\mathbf{d}} = \begin{Bmatrix} \dot{\mathbf{u}} \\ \omega \end{Bmatrix} \tag{9.1}$$

$$\Phi(\mathbf{d}, \dot{\mathbf{d}}) = 0 \tag{9.2}$$

where $\mathbf{M}$ is the system mass matrix, $\dot{\mathbf{d}}$ is the generalized velocity vector, $\mathbf{u}$ is the translational degrees of freedom, ω is the angular velocity vector, $\mathbf{B} = \partial\Phi/\partial\mathbf{d}$ is the constraint projection matrix, λ is the constraint force vector, Φ are the constraint conditions that are imposed either on the subsystem boundaries or on the kinematical relations among the generalized coordinates, t is the time, $(\dot{\ })$ denotes time differentiation, and $\mathbf{Q}$ is the generalized applied force plus non-linear inertia forces.

We observe from (9.1) and (9.2) that the task for solving the governing multibody dynamical equations constitutes three computational procedures: accurate computations of the constraint force, λ, or their elimination from the equations of motion (9.1), updates of the angular orientations, and the direct integration of the translational displacement $\mathbf{u}$. To this end, we will first examine two distinctive ways of handling λ.

9.2.2 *Techniques for handling constraint conditions*

In principle, it is better to eliminate the constraint conditions, if possible, if the corresponding forces are not needed for design or interface with other analysis modules. For example, if the system consists of open-tree configurations and no active controller is applied, then it is best to eliminate the joint constraint attributes. On the other hand, when the system includes multiple closed-loop configurations or active controllers are present on several joints, then it becomes important to compute the Lagrange multipliers as accurately as possible.

First, one can easily eliminate the system constraint forces via a coordinate partitioning strategy whenever any or all of the system components possess an open-tree topology. In the second procedure, we present a stabilization procedure for solving the Lagrange multipliers. A distinct feature of this stabilization procedure is that it can be implemented in a stand-alone module, thus can be interfaced not only with the equation solver for rigid-body systems but with that for flexible-body systems as well.

Parallel implementation of coordinate partitioning technique

In this technique, a projection matrix that spans the null space of the constraint Jacobian matrix Φ_u is first constructed (see, e.g., Wehage and Haug [9]). A parallel methodology (Chiou [10]; Chiou *et al.* [11]) based on an arrowhead algorithm then can be applied to the resulting complementary set of equations of motion. We will present the procedure for open-tree systems. For a system that contains closed loops, a cut-joint technique can be used so that the present scheme can be equally applied.

Let us introduce a projection matrix $\mathbf{A}$ such that, when its transposed matrix acts on the constraint force $\mathbf{B}^T \lambda$, it gives

$$\mathrm{A}^T \mathrm{B}^T \lambda = 0. \tag{9.3}$$

This projection matrix can be obtained by expressing the total generalized velocity $\dot{\mathbf{d}}^T = \langle \dot{\mathrm{u}}^T \omega^T \rangle$ in terms of the independent velocities $\dot{\mathbf{d}}^I$ and their time derivatives as

$$\dot{\mathbf{d}} = \mathbf{A}\dot{\mathbf{d}}^I, \qquad \ddot{\mathbf{d}} = \mathbf{A}\ddot{\mathbf{d}}^I + \dot{\mathbf{A}}\dot{\mathbf{d}}^I. \tag{9.4}$$

Due to the property of (9.3), premultiplying the equations of motion (9.1) by A^T yields

$$\mathrm{A}^T \mathrm{M}\ddot{\mathrm{d}} = \mathrm{A}^T \mathrm{Q}. \tag{9.5}$$

In conventional procedure, $\ddot{\mathbf{d}}$ in the above equation is replaced by (9.4b) and $\ddot{\mathbf{d}}^I$ is then solved from the reduced equations of motion. In the solution to be described below. instead of solving the reduced equations of motion, we augment (9.4b) to (9.5) to form an arrowhead matrix equation:

$$\begin{bmatrix} \mathrm{M} & -\mathrm{MA} \\ -\mathrm{A}^T\mathrm{M} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{d}} \\ \ddot{\mathbf{d}}^I \end{Bmatrix} = \begin{Bmatrix} \mathrm{M}\dot{\mathrm{A}}\dot{\mathbf{d}}^I \\ -\mathrm{A}^T\mathrm{Q} \end{Bmatrix} \tag{9.6}$$

which can be partitioned as

$$\begin{bmatrix} \mathbf{M}_{(1)} & 0 & 0 & 0 & \cdots & \mathbf{D}_{(1,n+1)} \\ 0 & \mathbf{M}_{(2)} & 0 & 0 & \cdots & \mathbf{D}_{(2,n+1)} \\ 0 & 0 & \mathbf{M}_{(3)} & 0 & \cdots & \mathbf{D}_{(3,n+1)} \\ 0 & 0 & 0 & \vdots & \cdots & \mathbf{D}_{(4,n+1)} \\ \vdots & \vdots & \vdots & \vdots & \mathbf{M}_{(n)} & \vdots \\ \mathbf{D}_{(n+1,1)} & \mathbf{D}_{(n+1,2)} & \mathbf{D}_{(n+1,3)} & \cdots & \cdots & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{d}}_1 \\ \ddot{\mathbf{d}}_2 \\ \ddot{\mathbf{d}}_3 \\ \vdots \\ \ddot{\mathbf{d}}_n \\ \ddot{\mathbf{d}}^f \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \\ \vdots \\ \mathbf{g}_n \\ \mathbf{g}^f \end{Bmatrix} \tag{9.7}$$

where n is the total number of bodies in the system. Decomposed in a manner convenient for parallel computations, one obtains

$$\begin{aligned} &\mathbf{M}_j\ddot{\mathbf{d}}_j + \mathbf{D}_{(j,\,n+1)}\ddot{\mathbf{d}}^f = \mathbf{g}_j, \qquad j = 1, \ldots, n \\ &\sum_{j=1}^{n} \mathbf{D}_{(n+1,\,j)}\ddot{\mathbf{d}}_j = \mathbf{g}^f \end{aligned} \tag{9.8}$$

where

$$\sum_{j=1}^{n} \mathbf{D}_{(n+1,\,j)} = -\sum_{j=1}^{n} \mathbf{A}_j^{\mathrm{T}}\mathbf{M}_j, \quad \mathbf{D}_{(j,\,n+1)} = -\mathbf{M}_j\mathbf{A}_j, \qquad j = 1, \ldots, n$$

$$\mathbf{g}_j = (\mathbf{M}\dot{\mathbf{A}}\dot{\mathbf{d}}^f)_j, \quad j = 1, \ldots, n, \qquad \mathbf{g}^f = -\sum_{j=1}^{n} \mathbf{A}^{\mathrm{T}}\mathbf{Q}_j.$$

Each diagonal submatrix $\mathbf{M}_j$ represents the local mass matrix which is decoupled and can be factorized concurrently. An off-diagonal submatrix $\mathbf{D}_{(j,n+1)}$ denotes the coupling between connecting bodies in the system. Since $\mathbf{M}$ is a constant matrix, (9.8) becomes

$$\ddot{\mathbf{d}}_j = \mathbf{M}_j^{-1}(\mathbf{D}_{(j,\,n+1)}\ddot{\mathbf{d}}^f - \mathbf{g}_j). \tag{9.9}$$

Substituting (9.9) into (9.8b) gives a form of *Schur complement*:

$$\sum_{j=1}^{n} \mathbf{D}_{(n+1,\,j)}\mathbf{M}_j^{-1}\mathbf{D}_{(j,\,n+1)}\ddot{\mathbf{d}}^f = \sum_{j=1}^{n} \mathbf{D}_{(n+1,\,j)}\mathbf{M}_j^{-1}\mathbf{Q}_j - \sum_{j=1}^{n} \mathbf{D}_{(n+1,\,j)}\dot{\mathbf{A}}\dot{\mathbf{d}}^f. \tag{9.10}$$

The preceding treatment of the reduced equations of motion provides several parallel computational features. First, the parallelism can be exploited by mapping each processor onto a group of bodies so that independent computations such as the left-hand side of (9.10) can be performed concurrently. Second, since $\mathbf{M}$ is a constant mass matrix, it needs to be factored only once. Third, to solve for $\ddot{\mathbf{d}}^f$, a parallel sparse solver may be utilized. Finally, once $\ddot{\mathbf{d}}^f$ is obtained, computations of $\dot{\mathbf{d}}$ from (9.4) is also easily parallelizable.

Stabilization of constraint violations

When the Lagrange multipliers cannot be eliminated or are to be retained for other purposes, one must solve for them. It has been known for some time that a straightforward direct time integration of the governing differential equation (9.1), augmented with a linearized form of the system constraints (9.2), often incur unacceptably high errors in the numerical solution. Of several techniques proposed to date, perhaps the method proposed by Baumgarte [12, 13] is the earliest known stabilization technique for computing the constraint forces.

While the method due to Baumgarte works relatively well, it requires an *a priori* determination of stabilization parameters and the method breaks down when the number of independent system constraints change due to varying configurations. To cope with the varying system constraints without experiencing singularities, a penalty-based stabilization has been developed in Park and Chiou [6]. The penalty procedure recasts the constraint equation in the form

$$\boldsymbol{\lambda} = \frac{1}{\varepsilon}\boldsymbol{\Phi} \tag{9.11}$$

as the basic constraint equations instead of (9.2) for both the holonomic and non-holonomic constraint conditions. We then time-differentiate, for the holonomic case, the above penalty-based equation to obtain:

$$\dot{\boldsymbol{\lambda}} = \frac{1}{\varepsilon}\mathbf{B}\,\dot{\mathbf{d}}, \qquad \dot{\mathbf{d}} = \begin{Bmatrix} \dot{\mathbf{u}} \\ \boldsymbol{\omega} \end{Bmatrix}. \tag{9.12}$$

The numerical solution to the above companion differential equation is effected as follows.

The constrained equation of motion (9.1) is integrated once using the implicit integration rule

$$\dot{\mathbf{d}}^{n+1/2} = \dot{\mathbf{d}}^{n} + \delta\,\ddot{\mathbf{d}}^{n+1/2}, \qquad \delta = \frac{h}{2}$$

to yield

$$\dot{\mathbf{d}}^{n+1/2} = \delta\,\mathbf{M}^{-1}(\mathbf{Q}^{n+1/2} - \mathbf{B}^{\mathrm{T}}\boldsymbol{\lambda}^{n+1/2}) + \dot{\mathbf{d}}^{n}. \tag{9.13}$$

This expression is substituted into (9.12) to obtain the stabilized differential equation for the Lagrange multipliers

$$\varepsilon\dot{\lambda}^{n+1/2} + \delta \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^{\mathrm{T}}\lambda^{n+1/2} = \delta \mathbf{B}\mathbf{M}^{-1}\mathbf{Q}^{n+1/2} + \mathbf{B}\dot{\mathbf{d}}^{n}.$$

When the above equation is integrated once more with the trapezoidal rule, we obtain the following discrete update for λ:

$$(\varepsilon I + \delta^2 \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^{\mathrm{T}})\lambda^{n+1/2} = \varepsilon\lambda^{n} + \mathbf{r}_{\lambda}^{n+1/2} \tag{9.14}$$

$$\mathbf{r}_{\lambda}^{n+1/2} = \delta^2 \mathbf{B}\mathbf{M}^{-1}\mathbf{Q}^{n+1/2} + \delta\mathbf{B}\dot{\mathbf{d}}^{n}. \tag{9.15}$$

The solution procedure for λ presented above can now be invoked in a staggered manner in conjuction with the generalized coordinate solver to be described below.

9.2.3 *Time integration of MBD equations of motion*

Once λ is computed by the procedure in Section 9.2.2 or $\ddot{\mathbf{d}}$ when using the partitioning algorithm, one still needs to compute $\dot{\mathbf{d}}$, u and the angular orientations and their parameters at each time step. This task is carried out by employing an explicit–implicit transient analysis algorithm that exploits the special kinematical relationships of the generalized rotational coordinates vs. the angular velocity, namely, the Euler parameters. First, the integration of the translational coordinates and the angular velocity is accomplished by the central difference formula. It should be mentioned that the use of the central difference formula does impose a step-size restriction due to its stability limit ($\omega_{\max} h \leq 2$) where $\omega_{\max}$ is the highest angular velocity of the system components for rigid-body systems or the highest frequency of the entire flexible members for flexible-body systems. The simplicity of its programming effort and robustness of its solution results can often become compelling enough to adopt an explicit formula, which is the view taken here.

Explicit translational coordinate integrator

In the conventional structural dynamics analysis, explicit time integration of the equations of motion by the central difference formula involves the following two updates per step:

$$\begin{aligned} \dot{\mathbf{u}}^{n+1/2} &= \dot{\mathbf{u}}^{n-1/2} + h\,\ddot{\mathbf{u}}^{n} \\ \mathbf{u}^{n+1} &= \mathbf{u}^{n} + h\,\dot{\mathbf{u}}^{n+1/2}. \end{aligned} \tag{9.16}$$

Unfortunately, the same integrator is not directly applicable to the rotational part of the equations of motion since $\boldsymbol{\omega}$ is not directly integrable to yield the total rotational quantities except for some special kinematic configurations. This motivates us to partition $\dot{\mathbf{d}}$ into the translational velocity vector, $\dot{\mathbf{u}}$, which is directly integrable and the angular velocity vector, $\boldsymbol{\omega}$, which is not, and treat them differently, viz.:

$$\ddot{\mathbf{d}} = \begin{Bmatrix} \ddot{\mathbf{u}} \\ \dot{\omega} \end{Bmatrix}, \qquad \dot{\mathbf{d}} = \begin{Bmatrix} \dot{\mathbf{u}} \\ \omega \end{Bmatrix}. \tag{9.17}$$

The equations of motion can be written according to the above partitioning as

$$\begin{bmatrix} \mathbf{M}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_\omega \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \dot{\omega} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q}_u \\ \mathbf{Q}_\omega \end{Bmatrix} \tag{9.18}$$

where

$$\begin{Bmatrix} \mathbf{Q}_u \\ \mathbf{Q}_\omega \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_u - \mathbf{D}_u(\dot{\mathbf{u}}) - \mathbf{S}_u(\mathbf{u}, \mathbf{q}) - \mathbf{B}_u^T \lambda \\ \mathbf{f}_\omega - \mathbf{D}_\omega(\omega) - \mathbf{S}_\omega(\mathbf{u}, \mathbf{q}) - \mathbf{B}_\omega^T \lambda \end{Bmatrix} \tag{9.19}$$

in which the subscripts $(\mathbf{u}, \omega)$ refer to the translational and the rotational motions, respectively, $\mathbf{f}$ is the external force vector, $\mathbf{D}$ is the generalized damping force including the centrifugal force, $\mathbf{S}$ is the internal force vector including member flexibility, $\mathbf{q}$ is the angular orientation parameters, $\mathbf{B}_u$ and $\mathbf{B}_\omega$ are the partition of the combined gradient matrices of the constraint conditions (9.2).

First, assume that $\mathbf{u}^{n+1/2}$ and $\mathbf{q}^{n+1/2}$ are already computed so that we can compute $\ddot{\mathbf{u}}^{n+1/2}$ and $\dot{\omega}^{n+1/2}$:

$$\begin{Bmatrix} \ddot{\mathbf{u}}^{n+1/2} \\ \dot{\omega}^{n+1/2} \end{Bmatrix} = \mathbf{M}^{-1} \begin{Bmatrix} \mathbf{Q}_u \\ \mathbf{Q}_\omega \end{Bmatrix}. \tag{9.20}$$

Second, update the translational velocity at the step $(n+1)$ by

$$\dot{\mathbf{u}}^{n+1} = \dot{\mathbf{u}}^{n} + h\,\ddot{\mathbf{u}}^{n+1/2}. \tag{9.21}$$

Third, we update the translational displacement, $\mathbf{u}$, by

$$\mathbf{u}^{n+3/2} = \mathbf{u}^{n+1/2} + h\,\dot{\mathbf{u}}^{n+1}. \tag{9.22}$$

The updating of the angular orientations must be treated with care, and is described below.

Updating of angular velocity via discrete angular momentum conservation

In order to update the angular velocity and angular orientations, we combine judiciously a momentum-conserving form of the central difference algorithm and the mid-point implicit rule for computing the Euler parameters as follows (Park and Chiou [14]).

First, we retain the basic central difference formula for computing the angular velocity at the half steps:

$$\omega^{n+1/2} = \omega^{n-1/2} + h\dot{\omega}^{n} \tag{9.23}$$

where $\dot{\omega}$ is computed from the equations of motion, utilizing the angular velocity obtained from the discrete angular conservation law, as described shortly.

Second, $\omega^{n+1/2}$ is used to integrate the Euler parameters by

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\omega \\ \omega & -\tilde{\omega} \end{bmatrix}\mathbf{q} = \mathbf{A}(\omega)\,\mathbf{q} \tag{9.24}$$

where $\mathbf{q} = (q_0 q_1 q_2 q_3)^{\mathrm{T}}$ are the Euler parameters expressed in the body-fixed frame and $\tilde{\omega}$ denotes the skew-symmetric angular velocity tensor given by

$$\begin{bmatrix} & 0 & -\omega_3 & \omega_2 \\ \tilde{\omega} = & \omega_3 & 0 & -\omega \\ & -\omega_2 & \omega_2 & 0 \end{bmatrix}. \tag{9.25}$$

Implicit integration (9.24) by the mid-point rule yields

$$\begin{aligned} \mathbf{q}^{n+1} - \mathbf{q}^{n} &= h\dot{\mathbf{q}}^{n+1/2} = h\,\mathbf{A}(\omega^{n+1/2})\mathbf{q}^{n+1/2} \\ &= \frac{h}{2}\mathbf{A}(\omega^{n+1/2})\,(\mathbf{q}^{n+1} + \mathbf{q}^{n}) \end{aligned} \tag{9.26}$$

where $\mathbf{A}(\omega^{n+1/2})$ can be viewed as the tangent matrix at the mid-configuration whereas $\mathbf{q}^{n+1/2} = (\mathbf{q}^{n+1} + \mathbf{q}^{n})/2$ is the mid-point average value. It was shown in Park *et al.* [7] that $\mathbf{q}^{n+1}$ can be expressed as

$$\mathbf{q}^{n+1} = \frac{1}{\Delta}\left[\mathbf{I} + \frac{h}{2}\mathbf{A}(\omega^{n+1/2})\right]^{2}\mathbf{q}^{n} \tag{9.27}$$

where $\Delta = 1 + h^2(\omega_1^2 + \omega_2^2 + \omega_3^2)/4$.

The updated Euler parameters $\mathbf{q}^{n+1}$ are then normalized to satisfy the constraint condition

$$\mathbf{q}^{n+1}\cdot\mathbf{q}^{n+1} = 1. \tag{9.28}$$

Once $\mathbf{q}^{n+1}$ is available, the angular orientation at t^{n+1} is then obtained by

$$\mathbf{R}^{n+1} = (2q_0^2 - 1)\mathbf{I} + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^{\mathrm{T}} - 2q_0\tilde{\bar{\mathbf{q}}}, \qquad \bar{\mathbf{q}} = (q_1 q_2 q_3)^{\mathrm{T}} \tag{9.29}$$

where the superscript $(n + 1)$ is omitted on $\mathbf{q}$.

Third, the angular velocity ω^{n+1} that is needed to compute $\dot{\omega}$ for the next step integration is obtained via a discrete version of the angular momentum conservation law:

$$\mathbf{M}_\omega \omega^{n+1} - \mathbf{M}_\omega \omega^n = h\tau^{n+1/2} \tag{9.30}$$

where $\mathbf{M}_\omega$ is the moment of inertia, ω is the angular velocity vector, and τ is the applied moment, all expressed in body-fixed frame at the configuration t^k, $0 \leq k \leq$ n. For computational simplicity, we will choose $k = 0$, i.e. the initial configuration so that (9.30) becomes

$$(\mathbf{R}^{n+1})^T \mathbf{M}_\omega \omega^{n+1} - (\mathbf{R}^n)^T \mathbf{M}_\omega \omega^n = h(\mathbf{R}^{n+1/2})^T \tau^{n+1/2} \tag{9.31}$$

where the matrix **R** is the rotation transformation matrix from the inertial basis **e** to the body fixed configuration **b** according to

$$\mathbf{b}^n = \mathbf{R}^n \mathbf{e} \tag{9.32}$$

and $\mathbf{M}_\omega$, ω and τ are now expressed in the superscript-indexed discrete **b**-bases. Therefore, the discrete angular momentum equation (9.30) becomes

$$\omega^{n+1} = \mathbf{M}_\omega^{-1} \mathbf{R}^{n+1} \left((\mathbf{R}^n)^T \mathbf{M}_\omega \omega^n + h(\mathbf{R}^{n+1/2})^T \mathbf{Q}_\omega{}^{n+1/2} \right) \tag{9.33}$$

where τ is replaced by $\mathbf{Q}_\omega$ from the lefthand side of (9.18).

It is noted that, whereas the standard difference formula (9.16) satisfies the linear momentum conservation for constant and linearly varying $\mathbf{Q}_\omega$, (9.33) indicates that the use of a common basis is essential for the conservation of angular momentum. A similar approach was successfully utilized by Simo and Wong [15] in their development of a family of implicit algorithms.

Fourth, the angular acceleration needed for the next time step $(n + 1)$ is then computed for each rigid body by

$$\dot{\omega}^{n+1} = \mathbf{M}_\omega^{-1} (\mathbf{Q}^{n+1} - \tilde{\omega}^{n+1} \mathbf{M}_\omega \tilde{\omega}^{n+1}). \tag{9.34}$$

Equations 9.23)–(9.34) constitute the present modified central difference algorithm for integrating the rotational equations of motion in multibody systems. However, as many engineering multibody systems involve both holonomic and nonholonomic constraints, the computation of $\mathbf{Q}^{n+1/2}$ is not as straightforward as (9.18) implies.

For a momentum-conserving implicit algorithm, the reader may consult Simo and Wong [15]. For applications of the preceding MBD procedures to flexible multibody dynamics, one may refer to Downer [16], Downer *et al.* [17] and Downer and Park [18], who solve the flexible appendage deployment problem, among others. Finally, other MBD recent approaches can be found in Haug and Deyo [19].

9.3 ALGORITHMS FOR CONTROL-STRUCTURE INTERACTION SIMULATION

A second topic we should like to report in this survey is computational methods for the simulation of dynamic response of structures that are subject to active control forces. A general case of structural response under active control forces involves both large-angle rigid motions as well as transient flexible vibrations. Engineering examples include the manoeuvring of robotic arms, satellite attitude changes, deployment and vibration control of large space structures, and active vibration suppression of rotating machinery and vehicle suspension systems.

When relatively small size models are adequate for describing the predominant motions and vibrations, the resulting active control strategies also can consist of a small number of actuators and sensors. However, as the structural model needs to be large due to the physical nature of the problem or due to the high-precision requirement, so must be the size of the actuator/sensor numbers. It is for such large-scale control-structure problems that the following simulation methodology has been developed.

Specifically, simulation tasks for control-structure interaction (CSI) problems involve several computational elements and discipline-oriented models such as structural dynamics, control law synthesis, state estimation, actuator and sensor dynamics, thermal analysis, liquid sloshing and swirling, environmental disturbances, and manoeuvring thrusts and torques. Because each of these computational elements can be large, it is usually not practical to assemble these computational elements into a single set of equations of motion and perform the analysis in its totality, which will be referred to as the *simultaneous solution approach*. First, the equation size of the total system can be simply too large for many existing computers. Second, the simultaneous solution by treating the coupled interaction equations as one system may destroy the sparsity of the attendant matrices, thus requiring excessive computations and storage space. Most important of all, any changes in the model or in the computational procedures will engender significant modifications of the required analysis software modules and hence require a painstaking software verification effort.

In order to alleviate the aforementioned difficulties that exist in the *simultaneous solution approach*, a partitioned solution procedure that takes the following considerations into account has been developed. First, software development of any new capability is costly and time-consuming; thus, if at all possible, it is preferable to utilize existing single-field analysis modules to conduct the coupled-field interaction analysis. Second, the tasks for model generation and methods development of each field are best accomplished by relying on the experts of each single-field discipline. In order to accommodate both the software considerations and the single-field expertise, a *partitioned (or divide-and-conquer) analysis procedure* has been developed for control-structure interaction analysis for direct output feedback systems (Belvin [20]; Park and Belvin [21]). The procedure abandons the conventional way of treating the CSI problems as one entity. Instead, it treats the structure (or plant), the observer, and the controller/observer interaction terms as separate entities. Thus, the CSI problem is recognized as a coupled-field problem and a divide-and-conquer strategy adopted for the development of a real-time computational procedure. It should be mentioned that a similar concept has been successfully applied to other interaction analyses such as fluid-structure interactions (Park *et al.* [22]), multi-structural interaction systems (Park [23]; Felippa and Park [3]; Park and Felippa [2]), earth dam and pore-fluid interactions (Park [23]; Zienkiewicz *et al.* [24] and multibody systems with constraints (Park *et al.* [7]; Chiou [10]; Downer [16]).

The *partitioned analysis procedure* hinges on two software and computational aspects. First, at each discrete time increment, the equations of motion for each discipline are solved separately by considering the interaction terms as external disturbances or applied forces. Second, when necessary, computational stabilization and accuracy improvements are introduced through augmentations and/or equation modifications. It is important to note that such partitioned solutions of each discipline equation can be carried out on either a sequential or a parallel machine if certain message passing and memory-conflict issues are handled appropriately.

9.3.1 *Equations of motion for control-structure interaction systems*

The discrete equations of motion for control-structure interaction systems may be described by [25]

$$\left.\begin{array}{lll}
\text{Structure:} & \text{(a)} & \mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \\
 & & \mathbf{q}(0) = \mathbf{q}_o, \qquad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \\
\text{Sensor output:} & \text{(b)} & \mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \\
\text{Estimator:} & \text{(c)} & \dot{\bar{\mathbf{x}}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{E}\mathbf{f} + \bar{\mathbf{B}}\mathbf{u} + \mathbf{L}_\gamma \\
 & & \bar{\mathbf{x}}(0) = 0 \\
\text{Control force:} & \text{(d)} & \mathbf{u} = -\mathbf{F}\bar{\mathbf{x}} \\
\text{Estimation error:} & \text{(e)} & \gamma = \mathbf{z} - (\mathbf{H}_d\bar{\mathbf{q}} + \mathbf{H}_v\dot{\bar{\mathbf{q}}})
\end{array}\right\} \qquad (9.35)$$

where

$$\mathbf{x} = \begin{Bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{Bmatrix}, \qquad \bar{\mathbf{x}} = \begin{Bmatrix} \bar{\mathbf{q}} \\ \dot{\bar{\mathbf{q}}} \end{Bmatrix}$$

and

$$\mathbf{H} = [\mathbf{H}_d \quad \mathbf{H}_v], \qquad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{bmatrix}, \qquad \mathbf{E} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} 0 & \mathrm{I} \\ -\mathrm{M}^{-1}\mathrm{K} & -\mathrm{M}^{-1}\mathrm{D} \end{bmatrix}, \qquad \bar{\mathbf{B}} = \begin{bmatrix} 0 \\ \mathrm{M}^{-1}\mathrm{B} \end{bmatrix}, \quad \mathbf{F} = [\mathbf{F}_1 \quad \mathbf{F}_2].$$

In the preceding equations, **M** is the mass matrix, **D** is the damping matrix, **K** is the stiffness matrix, $\mathbf{f}(t)$ is the applied force, **B** is the actuator location matrix, **G** is the disturbance location matrix, **q** is the generalized displacement vector, **w** is a disturbance vector and the superscript dot denotes time differentiation. In (9.35b), **z** is the measured sensor output. The matrix $\mathbf{H}_d$ is the matrix of displacement sensor locations and $\mathbf{H}_v$ is the matrix of velocity sensor locations. The vector **v** is measurement noise. The state estimator in (9.35c) is assumed to be based on either the Kalman filter (Kalman and Bucy [26]) or a Luenberger observer [27] if the system is deterministic. The superscript $^{-}$ denotes the estimated states. The actuator output, **u**, is a function of the state estimator variables, $\bar{\mathbf{q}}$ and $\dot{\bar{\mathbf{q}}}$, and $\mathbf{F}_1$ and $\mathbf{F}_2$ are control gains determined for example by pole–zero placement or from the solution of an optimal control problem. The observer is governed by L, the filter gain matrix. For the special case where L_1 is the null matrix (i.e. $\dot{\bar{\mathbf{q}}} = \bar{\dot{\mathbf{q}}}$), a second-order state estimator can be expressed as

$$\mathbf{M}\ddot{\bar{\mathbf{q}}} + \mathbf{D}\dot{\bar{\mathbf{q}}} + \mathbf{K}\bar{\mathbf{q}} = \mathbf{f} + \mathbf{Bu} + \mathbf{ML}_2\gamma. \tag{9.36}$$

The effect of the above simplification on the observer stability and convergence is discussed in detail in Belvin [20] and Belvin and Park [28].

9.3.2 *Simultaneous solution approach*

The numerical solution of (9.35) by the simultaneous solution approach begins with appropriate initial conditions, the feedback gain **F** and the filter gain **L**. The structure equation is written in first-order form

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Ef} + \bar{\mathbf{B}}\mathbf{u} + \bar{\mathbf{G}}\mathbf{w} \tag{9.37}$$

where

$$\bar{\mathrm{G}} = \begin{bmatrix} 0 \\ \mathrm{M}^{-1}\mathrm{G} \end{bmatrix}.$$

The control gains and observer gains can be synthesized independently by noting that the stability of the structural system and the observer error stability are uncoupled. Introducing the error equation by the deterministic form (9.35) as

$$\bar{\mathbf{e}} = \mathbf{x} - \bar{\mathbf{x}} = \begin{Bmatrix} \mathbf{q} - \bar{\mathbf{q}} \\ \dot{\mathbf{q}} - \dot{\bar{\mathbf{q}}} \end{Bmatrix} \tag{9.38}$$

and eliminating **u** yields

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \dot{\bar{\mathbf{e}}} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} - \bar{\mathbf{B}}\mathbf{F} & \bar{\mathbf{B}}\mathbf{F} \\ \mathbf{0} & \mathbf{A} - \mathbf{L}\mathbf{H} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \bar{\mathbf{e}} \end{Bmatrix} + \begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix} \mathbf{f} + \begin{bmatrix} \bar{\mathbf{G}} \\ \mathbf{0} \end{bmatrix} \mathbf{w}. \tag{9.39}$$

The stability of (9.39) is governed by the stability of $[\mathbf{A} - \bar{\mathbf{B}}\mathbf{F}]$ and $[\mathbf{A} - \mathbf{L}\mathbf{H}]$. Thus, the control gain **F** is suitably chosen from the matrix $[\mathbf{A} - \bar{\mathbf{B}}\mathbf{F}]$ and the observer gain **L** from the matrix $[\mathbf{A} - \mathbf{L}\mathbf{H}]$.

Subsequently, the simultaneous solution approach eliminates **u** and **z** from (9.35a, c) and then solves the observer based closed-loop equations

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \dot{\bar{\mathbf{x}}} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & -\bar{\mathbf{B}}\mathbf{F} \\ \mathbf{L}\mathbf{H} & \mathbf{A} - \bar{\mathbf{B}}\mathbf{F} - \mathbf{L}\mathbf{H} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \tilde{\mathbf{x}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{E} \\ \mathbf{E} \end{Bmatrix} \mathbf{f} + \begin{Bmatrix} \bar{\mathbf{G}} \\ \mathbf{0} \end{Bmatrix} \mathbf{w}. \tag{9.40}$$

The embedding effects of both the controller and the state observer result in an unsymmetric and non-sparse system matrix of dimension ($4N$ by $4N$), where N is the number of structural degrees of freedom. Solution of (9.40) would require considerable software modifications of existing structural dynamics analysis programs for large-scale CSI simulation purposes. In addition to losing the computational advantages associated with the finite element based CSI equation, the simultaneous solution approach requires the control law to be embedded into the observer model. If the control law includes actuator, sensor and/or controller dynamics, additional states must be added to the observer. This greatly complicates the observer model and requires significant software development for each class of control law dynamics. The difficulties associated with the *simultaneous solution approach* have prompted development of *a partitioned solution approach* for the CSI equations as described below.

9.3.3 *Stabilization for computations of control force and estimation error*

The partitioned solution procedure numerically integrates the structural equations of motion (9.35a) and the observer equation (9.35c) by treating the control force **u** and the estimation error γ as if they were applied terms in the right-hand sides. In this way, simulation of control-structure interaction systems using the partitioned solution procedure can be carried out by a judicious employment of three software modules: the structural analyser to obtain **q**, the state estimator to obtain $\bar{\mathbf{q}}$, and the stabilized solver for the control force **u** and the state estimation error γ. Thus the partitioned procedure becomes computationally efficient and can preserve software modularity by exploiting the symmetric matrix form on the left-hand sides of (9.35a) and (9.35c).

However, computations of the control force **u** and the state estimation error γ by (9.35d) and (9.35e), respectively, can not only lead to an accumulation of errors but often can give rise to numerical instability. Hence, in order to make the partitioned solution procedure robust, it is imperative to stabilize the partitioned solution process and/or numerically filter the solution errors in computing **u** and γ. This is addressed below.

First, we time-differentiate (9.35c) to obtain

$$\dot{\mathbf{u}} = \mathbf{F}_1\dot{\bar{\mathbf{q}}} - \mathbf{F}_2\ddot{\bar{\mathbf{q}}}. \tag{9.41}$$

Substituting $\ddot{\bar{\mathbf{q}}}$ from (9.36) into the above equation, one obtains

$$\dot{\mathbf{u}} + \mathbf{F}_2\mathbf{M}^{-1}\mathbf{B}\mathbf{u} = -\mathbf{F}_2(\mathbf{M}^{-1}\dot{\bar{\mathbf{p}}} + \mathbf{L}_2\gamma) - \mathbf{F}_1\dot{\bar{\mathbf{q}}} \tag{9.42}$$

where the generalized rate of momentum $\dot{\bar{\mathbf{p}}}$ is given by

$$\dot{\bar{\mathbf{p}}} = (\mathbf{f} - \mathbf{D}\dot{\bar{\mathbf{q}}} - \mathbf{K}\bar{\mathbf{q}}). \tag{9.43}$$

The parabolic stabilization that led to equation (9.42) for computing the control law is sometimes called an equation augmentation procedure as it has not altered any part of the basic governing equation (9.35) except one time-differentiation of **u** assuming $\dot{\mathbf{u}}$ exists. However, this assumption is later removed through time discretization as will be shown later in the chapter.

It is noted that the homogenous part of (9.42) has the filtering effect of the form $(s\mathbf{I} + \mathbf{F}_2\mathbf{M}^{-1}\mathbf{B})^{-1}$ in parlance of classical control theory, where s is the Laplace transform operator, thus achieving the required stabilization. From the computational viewpoint, although $\mathbf{F}_2\mathbf{M}^{-1}\mathbf{B}$ is in general a full matrix, its size is relatively small, as the size of **u** is proportional to the number of actuators placed on the structure.

Similarly, for the observer estimation error γ one can stabilize its computation first by time-differentiating it

$$\dot{\gamma} + \mathbf{H}_v\mathbf{L}_2\gamma = \dot{\mathbf{z}} - (\mathbf{H}_d\dot{\bar{\mathbf{p}}} + \mathbf{H}_v\ddot{\bar{\mathbf{q}}}). \tag{9.44}$$

and substituting the observer equation into the above to obtain an augmented form of the observer error equation:

$$\dot{\gamma}\,\mathbf{H}_v\mathbf{L}_2\gamma = \dot{\mathbf{z}} - \mathbf{H}_v\mathbf{M}^{-1}(\dot{\bar{\mathbf{p}}} + \mathbf{B}\mathbf{u}) - \mathbf{H}_d\,\dot{\bar{\mathbf{q}}}. \tag{9.45}$$

9.3.4 *Stabilized partitioned equations and solution process*

The adoption of the second-order observer and the preceding stabilization thus replaces (9.35c), (9.35d) and (9.35e) by (9.36), (9.43) and (9.45), respectively, as summarized below.

$$\begin{array}{lll}
\text{Structure:} & \text{(a)} & \mathbf{M\ddot{q} + D\dot{q} + Kq = f + Bu + Gw} \\
 & & \mathbf{q}(0) = \mathbf{q}_0, \quad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \\
\text{Sensor output:} & \text{(b)} & \mathbf{z = Hx + v} \\
\text{Estimator:} & \text{(c)} & \mathbf{M\ddot{\bar{q}} + D\dot{\bar{q}} + K\bar{q} = f + Bu + ML_2}\gamma \\
 & & \bar{\mathbf{q}}(0) = 0, \quad \dot{\bar{\mathbf{q}}}(0) = 0 \\
\text{Control force:} & \text{(d)} & \mathbf{\dot{u} + F_2M^{-1}Bu} = -\mathbf{F}_2(\mathbf{M}^{-1}\dot{\bar{\mathbf{p}}} + \mathbf{L}_2\gamma) - \mathbf{F}_1\dot{\bar{\mathbf{q}}} \\
\text{Estimation error:} & \text{(e)} & \dot{\gamma} + \mathbf{H}_v\mathbf{L}_2\gamma = \dot{\mathbf{z}} - \mathbf{H}_v\mathbf{M}^{-1}(\dot{\bar{\mathbf{p}}} + \mathbf{Bu}) - \mathbf{H}_d\dot{\bar{\mathbf{q}}}
\end{array} \tag{9.46}$$

Note that the difference between the original governing equation set (9.35) and the above stabilized set (9.46) is an obstacle to computation of the control forces and the state estimation error vector.

9.3.5 *Stability and accuracy of partitioned solution procedure*

Computational stability analysis of partitioned procedures for a general coupled system is still in an evolving stage. Hence, the analysis herein applies the relevant results from (Belvin [20]; Park and Belvin [21]) in the present stability analysis of the partitioned CSI solution procedure. The partitioned CSI solution procedure presented in (9.46), even when discretized by unconditionally stable implicit time integration formulae, may still suffer from computational instability as it involves extrapolations to obtain $\mathbf{u}^{n+1/2}$ and $\gamma^{n+1/2}$. A complete stability analysis of the partitioned solution procedure for the coupled structural dynamics, observer and controller equations is difficult to perform unless the observer characteristics **H**, **L** and the controller characteristics **B**, **F** are specified. Hence, the analysis that follows is restricted to an ideal observer, i.e. $\gamma = 0$. In what follows, it is assumed that all of the stabilized equation set is time-discretized by a mid-point version of the trapezoidal rule.

In order to assess the computational stability of the present partitioned solution procedure, we construct a model single degree-of-freedom interaction equation as follows. First, neglecting structural damping a modal structural equation of motion can be expressed as

$$\ddot{y} + \omega^2 y = -u \tag{9.47}$$

where y is a generalized coordinate and ω is its associated frequency.

Second, the model controller is assumed to consist of both the position and velocity feedback with appropriate weights given by

$$u = \eta\omega_c^2 y + \zeta\omega_c\dot{y}, \qquad \omega_{\min} \leq \omega_c \leq \omega_{\max} \tag{9.48}$$

where ω_c is the feedback frequency, which ranges from the minimum to the maximum of the structural frequency contents, and η and ζ are positive scalar coefficients that signify the strength of the position and the velocity feedback, respectively.

Combining (9.47) with the stabilized form of (9.48) we have the model interaction equation as

$$\begin{aligned} &\ddot{y} + \omega^2 y = -u \\ &\dot{u} + \zeta\omega_c u = \eta\omega_c^2\dot{y} - \zeta\omega_c\omega^2 y. \end{aligned} \tag{9.49}$$

Thus, the model interaction equations given by (9.49) represent the case of full state feedback. They do not, however, reflect the mode-to-mode coupling that can occur in reduced-order feedback controller. Nevertheless, an analysis of the computational stability using the above model interaction equations should shed insight on the overall stability of the present partitioned solution procedure.

Time integration of the above model problem (9.49) by the mid-point rule

$$\begin{aligned} x^{n+1/2} &= x^n + \delta\dot{x}^{n+1/2} \\ \dot{x}^{n+1/2} &= \dot{x}^n + \delta\ddot{x}^{n+1/2} \\ x^{n+1} &= 2x^{n+1/2} - x^n \end{aligned} \tag{9.50}$$

with $\gamma = 0$ yields

$$\begin{aligned} &y_p^{n+1/2} = y^n + \delta\dot{y}^n \\ &(1 + \delta\zeta\omega_c)u_p^{n+1/2} = (\eta\omega_c^2 - \delta\zeta\omega_c\omega^2)\,y_p^{n+1/2} + \zeta\omega_c\,\dot{y}^n \\ &(1 + \delta^2\omega^2)y^{n+1/2} = -\,\delta^2 u^{n+1/2} + y^n + \delta\dot{y}^n \\ &y^{n+1} = 2y^{n+1/2} - y^n, \quad \dot{y}^{n+1/2} = (y^{n+1/2} - y^n)/\delta, \quad \dot{y}^{n+1} = 2\dot{y}^{n+1/2} - \dot{y}^n \\ &u^{n+1} = \eta\omega_c^2 y^{n+1} + \zeta\omega_c\dot{y}^{n+1} \end{aligned} \tag{9.51}$$

where $y_p^{n+1/2}$ is a stable predictor that is needed to initiate the staggered solution.

Computational stability of the above difference equation can be assessed by seeking a non-trivial solution in the form

$$\begin{Bmatrix} u^{n+1} \\ y^{n+1} \end{Bmatrix} = \lambda \begin{Bmatrix} u^n \\ y^n \end{Bmatrix} \tag{9.52}$$

such that

$$|\lambda| \leq 1 \tag{9.53}$$

for stability.

Substituting (9.52) into (9.51) and eliminating $\dot{y}$, one obtains

$$\mathbf{J}\begin{Bmatrix} u \\ y \end{Bmatrix}^n = 0 \tag{9.54}$$

where

$$\mathbf{J} = \begin{bmatrix} \delta(1 + \delta\zeta\omega_c)(\lambda + 1)^2 & 4\lambda(\delta^2\zeta\omega_c\omega^2 - \delta\eta\omega_c^2) + 2\zeta\omega_c(1 - \lambda) \\ \delta^2(\lambda + 1)^2 & (1 + \delta^2\omega^2)(\lambda + 1)^2 - 4\lambda \end{bmatrix}.$$

In order to test the stability requirement (9.53) on the characteristic equation, i.e., det|J| = 0, one transforms $|\lambda| \leq 1$ into the entire left-hand plane of the z-plane by

$$\lambda = \frac{1 + z}{1 - z}, \qquad |\lambda| \leq 1 \Leftrightarrow \mathrm{Re}(z) \leq 0. \tag{9.55}$$

Carrying out the necessary algebra we have from det|J(z)| = 0 the following z-polynomial equation:

$$(\delta^3 \zeta \omega_c \omega^2 - \delta^2 \eta \omega_c^2 + 1) z^2 + (\delta \zeta \omega_c) z + \delta^2 (\eta \omega_c^2 + \omega^2) = 0. \tag{9.56}$$

A test of the polynomial equation (9.56) for possible positive real roots by the Routh–Hurwitz criterion (Ganthmacher [29]) indicates that the partitioned procedure as applied to the model coupled equations (9.47) and (9.48) give a stable solution provided

$$(\delta^3 \zeta \omega_c \omega^2 - \delta^2 \eta \omega_c^2 + 1) \geq 0. \tag{9.57}$$

Note that, if there is no position feedback (i.e., $\eta = 0$), the model interaction equations solved by the present partitioned solution procedure (9.46) yields unconditionally stable solutions as (9.57) is automatically satisfied. Hence, a more critical stability assessment can be made by assuming no velocity feedback (i.e. $\zeta = 0$) for which we have for stability from (9.57)

$$h \leq \frac{2}{\sqrt{\eta}\omega_c}. \tag{9.58}$$

The preceding stability analysis on the model interaction equations permits us to make the following observations. First, equation (9.58) indicates that feedback frequency (ω_c) and the strength of the position feedback (η) dictate the computational stability and not the structural frequency (ω). In other words, the position feedback dictates the allowable step size for stability. Thus the highest frequency of the controller governs stability, not the highest frequency of the structure. Since most controllers are designed with reduced order structure models that ignore high frequency dynamics, the present solution procedure is not unduly restricted by stability. Second, if velocity feedback is present, the allowable step size for stability increases until $\zeta \geq \sqrt{(4\eta^3/27)}$, at which point the solution becomes unconditionally stable.

It should be noted that, instead of the stabilized form of control force equation (9.46d) or (9.49b), if the scalar form of (9.48) is used in the preceding stability analysis, the resulting stability limit is given by

$$h \leq \min\left(\frac{2}{\zeta\omega_c}, \ \frac{2\zeta}{\eta\omega_c}\right). \tag{9.59}$$

Assuming $\zeta \ll 1$, the first term in the above condition allows a sufficiently large step size. However, since $\zeta/\eta \approx 1$ for a balanced control law, it imposes a step-size restriction $h \approx 2/\omega_c$, which approaches the limit imposed on by a typical explicit integration formula. This proves the advantage of the present stabilized partitioned solution equation (9.46) solely from the computational stability viewpoint.

Although not elaborated herein, a stability analysis that includes an observer model and the state estimation error equation has been conducted with the following parameter choices:

$$\mathbf{L}_2 = [l_{21} \quad l_{22}], \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9.60}$$

in conjunction with the structural model and the controller model already used in (9.49). The analysis result yields the following step-size restriction:

$$h \leq \min\left(\frac{2}{\zeta\omega_c}, \frac{2}{\sqrt{\eta}\omega_c}, \frac{2}{\sqrt{l_{21}}}\right). \tag{9.61}$$

It should be noted that l_{21} corresponds to the Kalman filter gain magnitude which can be adjusted to be sufficiently small compared with ω^2 as can be assessed from equation (9.39b). Hence, provided $l_{21} < \omega_c$, the condition given by (9.58) is seen to govern the maximum stable step size by the present partitioned solution procedure.

For the general multidimensional case governed by (9.46), one observes that the stiffness proportional control force in practice reaches only a fraction of the total internal force ($\mathbf{u} = \eta\mathbf{K}\mathbf{q}, \quad \eta \ll 1$). Hence, even for a distributed stiffness proportional control configuration where $\omega_c \rightarrow \omega_{\max}$, the stable step size given by (9.58) should be much larger than the maximum stable step size of a typical explicit integration algorithm (say, $h_{\max} \leq 2/\omega_{\max}$). Therefore, the computational efficiency of the present partitioned solution procedure is established.

9.4 SOLUTION METHODS FOR COUPLED THERMAL–STRUCTURAL ANALYSIS

Coupled thermal–structural problems are becoming a major challenge in many engineering disciplines such as supersonic planes, satellites, superelectronic chips, and jet and combustion engines. Following the finite element formulations proposed by Wilson and Nickell [30], Nickell and Sackman [31], Oden [32], and Oden and Armstrong [33], among others, the semidiscrete coupled thermal–structural governing equations can be written as

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} - \mathbf{C}\boldsymbol{\theta} &= \mathbf{f} \\ \mathbf{Q}\dot{\boldsymbol{\theta}} + \mathbf{H}\boldsymbol{\theta} + \theta^o\mathbf{C}^T\dot{\mathbf{u}} &= \mathbf{r} \end{aligned} \tag{9.62}$$

where M, D and K are the mass, damping and stiffness matrices, and f is the prescribed structural loading vector; **Q**, **H** and **C** are the heat capacitance, heat diffusion and thermal expansion coupling matrices, and **r** is the external heat source, respectively; and θ_0 is the reference temperature.

9.4.1 *Conventional implicit solution procedures*

Suppose we are given two software modules, a structural analyser and a thermal conduction transient analysis module and are tasked to perform the coupled response analysis given by (9.62). The simplest way is then to move the coupling terms $\mathbf{C}\boldsymbol{\theta}$ and $\mathbf{C}^T\dot{\mathbf{u}}$ in the above equation to the right-hand sides and treat them as if they are an applied force and an additional source term, respectively. This will permit the use of two single disciplined-oriented software modules for the analysis of coupled problems. Computationally, this amounts to employing the following *staggered solution procedure*:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{u}}^{n+1/2} + \mathbf{D}\dot{\mathbf{u}}^{n+1/2} + \mathbf{K}\mathbf{u}^{n+1/2} &= \mathbf{f}^{n+1/2} + \mathbf{C}\boldsymbol{\theta}_p^{n+1/2} \\ \mathbf{Q}\boldsymbol{\theta}^{n+1/2} + \mathbf{H}\boldsymbol{\theta}^{n+1/2} &= \mathbf{r}^{n+1/2} - \theta_0\mathbf{C}^T\dot{\mathbf{u}}^{n+1/2} \end{aligned} \tag{9.63}$$

where $\boldsymbol{\theta}_p^{n+1/2}$ is the predicted temperature. It turns out that if $\mathbf{C}^T\dot{\mathbf{u}}^{n+1/2}$ is predicted instead of $\boldsymbol{\theta}^{n+1/2}$, one ends up with the same accuracy and stability limits (Park *et al.* [22]).

While the above implicit–implicit staggered procedure is simple to implement, it can be shown that it is only conditionally stable, even though the implicit integrators used to integrate the left-hand sides of (9.62) are *algorithmically* unconditionally stable. The stabilization procedure that we will describe is a mid-point rule modification of Farhat *et al.* [34]).

9.4.2 *Stabilization of implicit–implicit staggered solution procedure*

Stabilization of a general staggered solution procedure for coupled-field problems can be accomplished either by a differential-level stabilization and algebraic-level stabilization. In the past both stabilization strategies have been employed for fluid–structure problems, coupled pore fluid–soil interactions, and structure–structure interaction problems (Park *et al.* [22]; Park [23]; Felippa and Park [3]; Park and Felippa [2]).

In general one can stabilize the implicit–implicit procedure by modifying both or just one of the two field equations. A successful stabilization is the one that minimizes the impact of stabilization in terms of software modification and computational overhead. Of several stabilization strategies studied, a concurrent adaptation of both differential and algebraic-level augmentations was found to yield the most attractive staggered procedure. We now outline the stabilization process.

First, we employ a mid-point version of the trapezoidal rule as

$$
\begin{aligned}
\dot{\mathbf{y}}^{n+1/2} &= \dot{\mathbf{y}}^{n} + \delta\ddot{\mathbf{y}}^{n+1/2} \\
\mathbf{y}^{n+1/2} &= \mathbf{y}^{n} + \delta\dot{\mathbf{y}}^{n+1/2} \\
\mathbf{y}^{n+1} &= 2\mathbf{y}^{n+1/2} - \mathbf{y}^{n} \\
\dot{\mathbf{y}}^{n+1} &= 2\dot{\mathbf{y}}^{n+1/2} - \dot{\mathbf{y}}^{n}
\end{aligned}
\tag{9.64}
$$

where y can be either the displacement or the temperature vector in (9.62) and δ is one-half of the step size, $\delta = 1/2\Delta t$, and Δt is the time step size.

Second, time discretization of the thermal coupled equations (9.62b) to obtain

$$
(\mathbf{Q} + \delta\mathbf{H})\boldsymbol{\theta}^{n+1/2} = \delta\mathbf{r}^{n+1/2} + \mathbf{Q}\boldsymbol{\theta}^{n} - \delta\theta_0\mathbf{C}^{T}\dot{\mathbf{u}}^{n+1/2}. \tag{9.65}
$$

Note that in the above difference equation, the unknown structural coupling term is associated with the velocity $\dot{\mathbf{u}}^{n+1/2}$. It is this vector that has been found to play a key role in stabilization of the present procedure. In order to stabilize the extrapolation of the coupling term, we utilize an integrated form of $\dot{\mathbf{u}}^{n+1/2}$ from the structural equation (9.61a):

$$
\begin{aligned}
\dot{\mathbf{u}}^{n+1/2} &= \mathbf{B}\,[\mathbf{M}\dot{\mathbf{u}}^{n} + \delta(\mathbf{f}^{n+1/2} - \mathbf{K}\mathbf{u}^{n+1/2} + \mathbf{C}\boldsymbol{\theta}^{n+1/2})] \\
\mathbf{B} &= (\mathbf{M} + \delta\mathbf{D})^{-1}.
\end{aligned}
\tag{9.66}
$$

Upon substituting the above expression into (9.65), we obtain

$$
\begin{aligned}
\mathbf{G}\boldsymbol{\theta}^{n+1/2} &= \mathbf{R}^{n+1/2} + \delta^{2}\theta_0\mathbf{C}^{T}\mathbf{B}\mathbf{K}\mathbf{u}_p^{n+1/2} \\
\mathbf{R}^{n+1/2} &= \delta\mathbf{r}^{n+1/2} + \mathbf{Q}\boldsymbol{\theta}^{n} - \delta\theta_0\mathbf{C}^{T}\mathbf{B}(\mathbf{M}\dot{\mathbf{u}}^{n} + \delta\mathbf{f}^{n+1/2}) \\
\mathbf{G} &= \mathbf{Q} + \delta\mathbf{H} + \delta^{2}\theta_0\mathbf{C}^{T}\mathbf{B}\mathbf{C} \\
\mathbf{u}_p^{n+1/2} &= \mathbf{u}^{n}.
\end{aligned}
\tag{9.67}
$$

It is observed that the solution matrix **G** for the thermal equation is augmented with the additional matrix $\delta^{2}\theta_0\mathbf{C}^{T}\mathbf{B}\mathbf{C}$ and the prediction of $\dot{\mathbf{u}}_p^{n+1/2}$ is replaced by $\delta\mathbf{B}\mathbf{K}\mathbf{u}_p^{n+1/2}$. Also, note that the predictor for $\mathbf{u}_p^{n+1/2}$ is simply the previous step solution which has been found the most stable predictor when used in conjunction with the trapezoidal rule (Park [23]).

Once the thermal equation is stabilized as described above, the structural equation (9.62a) can be integrated in an existing structural analysis program as if the term $\mathbf{C}\boldsymbol{\theta}$ is an external force at each integration step. Time discretization of (9.62a) by the mid-point rule (9.64) yields

$$
\begin{aligned}
\mathbf{E}\mathbf{u}^{n+1/2} &= \mathbf{F}^{n+1/2} + \delta^{2}\mathbf{C}\boldsymbol{\theta}^{n+1/2} \\
\mathbf{E} &= \mathbf{M} + \delta\mathbf{D} + \delta^{2}\mathbf{K} \\
\mathbf{F}^{n+1/2} &= \delta^{2}\mathbf{f}^{n+1/2} + \mathbf{M}(\mathbf{u}^{n} + \delta\dot{\mathbf{u}}^{n}) + \delta\mathbf{D}\mathbf{u}^{n}.
\end{aligned}
\tag{9.68}
$$

The updating procedure for states at time step $(n + 1)$ is achieved as follows:

$$\begin{aligned}
\mathbf{u}_t^{n+1} &= 2\mathbf{u}^{n+1/2} - \mathbf{u}^n \\
\dot{\mathbf{u}}_t^{n+1} &= 2(\mathbf{u}^{n+1/2} - \mathbf{u}^n)/\delta - \dot{\mathbf{u}}^n \\
\boldsymbol{\theta}_t^{n+1} &= 2\boldsymbol{\theta}^{n+1/2} - \boldsymbol{\theta}^n \\
\ddot{\mathbf{u}}^{n+1} &= \mathbf{M}^{-1}(\mathbf{f}^{n+1} + \mathbf{C}\boldsymbol{\theta}_t^{n+1} - \mathbf{D}\dot{\mathbf{u}}_t^{n+1} - \mathbf{K}\mathbf{u}_r^{n+1}) \\
\dot{\mathbf{u}}^{n+1} &= \dot{\mathbf{u}}^n + \delta(\ddot{\mathbf{u}}^{n+1} + \ddot{\mathbf{u}}^n) \\
\mathbf{u}^{n+1} &= \mathbf{u}^n + \delta(\dot{\mathbf{u}}^{n+1} + \dot{\mathbf{u}}^n) \\
\dot{\boldsymbol{\theta}}^{n+1} &= \mathbf{Q}^{-1}(\mathbf{r}^{n+1} - \theta_0\mathbf{C}^{\mathrm{T}}\dot{\mathbf{u}}^{n+1} - \mathbf{H}\boldsymbol{\theta}_t^{n+1}) \\
\boldsymbol{\theta}^{n+1} &= \boldsymbol{\theta}^n + \delta(\dot{\boldsymbol{\theta}}^{n+1} + \dot{\boldsymbol{\theta}}^n).
\end{aligned} \tag{9.69}$$

9.4.3 *An analysis of stability and accuracy of stabilized procedure*

Stability of the staggered procedure presented in (9.67)–(9.69) can be assessed by adopting an analysis procedure, for example, outlined in Park [4]. First, we assume that the step-by-step numerical solution for a uniform step integration can be characterized by

$$\mathbf{y}^{n+1} = \lambda \mathbf{y}^n. \tag{9.70}$$

Hence, computational stability is maintained if

$$|\lambda| \le 1. \tag{9.71}$$

In order to invoke the well-known Routh–Hurwitz criterion, we map the stable zone, $|\lambda| \le 1$, onto the left-hand side of a z-plane by the following idempotent transformation:

$$\begin{bmatrix} \mathbf{u}^{n+1} \\ \dot{\mathbf{u}}^{n+1} \\ \ddot{\mathbf{u}}^{n+1} \\ \boldsymbol{\theta}^{n+1} \\ \dot{\boldsymbol{\theta}}^{n+1} \end{bmatrix} = \frac{1+z}{1-z} \begin{bmatrix} \mathbf{u}^{n} \\ \dot{\mathbf{u}}^{n} \\ \ddot{\mathbf{u}}^{n} \\ \boldsymbol{\theta}^{n} \\ \dot{\boldsymbol{\theta}}^{n} \end{bmatrix}. \tag{9.72}$$

Substitution of (9.70) and (9.72) into (9.67)–(9.69) with $\mathbf{D} = 0$ yields

$$\begin{bmatrix} z^2\mathbf{M} + \delta^2\mathbf{K} & -\delta^2\mathbf{C} \\ -(1 - z^2)\delta^2\theta_0\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{K} & z^2\mathbf{Q} + z\delta\mathbf{H} + \delta^2\theta_0\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{u}^n \\ \boldsymbol{\theta}^n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9.73}$$

whose characteristic equation is obtained from

$$\mathrm{Det}|\mathrm{M}z^3 + \mathrm{VM}\delta z^2 + \delta^2(\mathrm{K} + \theta_0\mathrm{CQ}^{-1}\mathrm{C}^{\mathrm{T}} + \delta^2\theta_0\mathrm{CQ}^{-1}\mathrm{C}^{\mathrm{T}}\mathrm{M}^{-1}\mathrm{K})z + \delta^3\mathrm{VK}| = 0 \qquad (9.74)$$

where

$$\mathrm{V} = \mathrm{CUC}^{\mathrm{T}}$$
$$\mathrm{U} = \mathrm{Q}^{-1}\mathrm{H}(\mathrm{C}^{\mathrm{T}}\mathrm{C})^{-1}.$$

The reader may find a complete stability analysis in Farhat *et al.* [34]. Hence, we offer the following synopsis. First, for a two-degree-of freedom problem, we have

$$\mathrm{M} = 1, \qquad \mathrm{K} = \omega^2, \qquad \mathrm{Q} = q, \qquad \mathrm{H} = h, \quad \mathrm{C} = c \qquad (9.75)$$

which, when substituted into the above characteristic equation, gives

$$a_3z^3 + a_2z^2 + a_1z + a_0 = 0 \qquad (9.76)$$

where

$$a_3 = 1, \qquad a_2 = \delta q, \qquad a_1 = \delta^2[\omega^2 + \frac{\theta_0 c^2}{qm}(1 + \delta^2\omega^2)], \qquad a_0 = \delta^3 q\omega^2.$$

Since δ, h, q, ω^2, θ_0, c^2, and $m \geq 0$, all the coefficients of the polynomial (9.76) in z are positive. Moreover, the quantity

$$a_1a_2 - a_0a_3 = \theta_0 hc^2\delta^3 mq^2(1 + \delta^2\omega^2)$$

is also positive, which demonstrates that the stabilized staggered solution procedure is unconditionally stable for the 2-d.o.f. model problem.

For multi-dimensional cases, the limiting case of $\mathrm{K} = 0$, which gives rise to a quadratically growing structural response due to thermal coupling, can be used as a pathological test:

$$|\mathrm{M}z^2 + \delta\mathrm{VM}z + \delta^2\theta_0\mathrm{CQ}^{-1}\mathrm{C}^{\mathrm{T}}| = 0. \qquad (9.77)$$

Since M is positive definite and the other two matrices are at least semi-definite, the stabilized staggered procedure for this limiting case is unconditionally stable via Bellman's theorem [35] as successfully utilized in Park [23]. Hence, we conclude that the procedure given by (9.67)–(9.69) is unconditionally stable.

As for accuracy, it can be shown that the stabilized staggered procedure is second-order accurate. This can be done first by expanding the difference equations (9.67)–(9.69) and using the governing coupled semidiscrete equations (9.62) where needed.

9.4.4 *Computational sequence*

When we ignore structural damping, diagonal structural mass and diagonal capacitance matrix, the stabilized computational procedure can be summarized as follows:

$$\mathbf{R}^{n+1/2} = \delta \mathbf{r}^{n+1/2} + \mathbf{Q}\boldsymbol{\theta}^{n} - \delta\theta_0 \mathbf{C}^{T}\mathbf{B}[\dot{\mathbf{u}}^{n} + \delta \mathbf{M}^{-1}(\mathbf{f}^{n+1/2} - \mathbf{K}\mathbf{u}^{n})] \tag{9.78}$$

$$(\mathbf{Q} + \delta \mathbf{H} + \delta^{2}\theta_0 \mathbf{C}^{T}\mathbf{M}^{-1}\mathbf{C})\boldsymbol{\theta}^{n+1/2} = \mathbf{R}^{n+1/2} \tag{9.79}$$

$$\mathbf{F}^{n+1/2} = \delta^{2}\mathbf{f}^{n+1/2} + \mathbf{M}(\mathbf{u}^{n} + \delta\dot{\mathbf{u}}^{n}) + \delta \mathbf{D}\mathbf{u}^{n} + \delta^{2}\mathbf{C}\boldsymbol{\theta}^{n+1/2} \tag{9.80}$$

$$(\mathbf{M} + \delta^{2}\mathbf{K})\mathbf{u}^{n+1/2} = \mathbf{F}^{n+1/2} \tag{9.81}$$

$$\begin{aligned}
\mathbf{u}_i^{n+1} &= 2\mathbf{u}^{n+1/2} - \mathbf{u}^{n} \\
\boldsymbol{\theta}_i^{n+1} &= 2\boldsymbol{\theta}^{n+1/2} - \boldsymbol{\theta}^{n} \\
\ddot{\mathbf{u}}^{n+1} &= \mathbf{M}^{-1}(\mathbf{f}^{n+1} + \mathbf{C}\boldsymbol{\theta}_i^{n+1} - \mathbf{K}\mathbf{u}_i^{n+1}) \\
\dot{\mathbf{u}}^{n+1} &= \dot{\mathbf{u}}^{n} + \delta(\ddot{\mathbf{u}}^{n+1} + \ddot{\mathbf{u}}^{n}) \\
\mathbf{u}^{n+1} &= \mathbf{u}^{n} + \delta(\dot{\mathbf{u}}^{n+1} + \dot{\mathbf{u}}^{n}) \\
\dot{\boldsymbol{\theta}}^{n+1} &= \mathbf{Q}^{-1}(\mathbf{r}^{n+1} - \theta_0 \mathbf{C}^{T}\dot{\mathbf{u}}^{n+1} - \mathbf{H}\dot{\boldsymbol{\theta}}^{n+1}) \\
\boldsymbol{\theta}^{n+1} &= \boldsymbol{\theta}^{n} + \delta(\dot{\boldsymbol{\theta}}^{n+1} + \dot{\boldsymbol{\theta}}^{n}).
\end{aligned} \tag{9.82}$$

The key for the efficiency of the above procedure compared with other candidate procedures is to utilize the matrix $\mathbf{C}^{T}\mathbf{M}^{-1}\mathbf{C}$, which appears in the left-hand side of (9.79) and is a symmetric banded matrix. Other possible stabilization involves $\mathbf{C}\mathbf{Q}^{-1}\mathbf{C}^{T}$ into the left-hand side of (9.79), which has much larger bandwidth than the former.

Some two-dimensional solutions of the thermal–structural interaction problems based on the above procedure are reported in Farhat *et al.* [34].

9.5 APPLICATION EXAMPLES

In the preceding sections three computational methods for performing coupled-field dynamics analyses have been surveyed. It is anticipated that as the analyst demands more realistic models, all the single-field components, viz., structures, control, thermal and multibody systems, may have to be included in a typical analysis. An example would be a satellite undergoing solar panel deployment as well as attitude stabilization and vibration control. Two examples that we include are a scenario of shuttle-based assembly of the space station for one-cargo segment, and a vibration control of a generic Earth-observing platform when it is subject to a reboosting thrust.

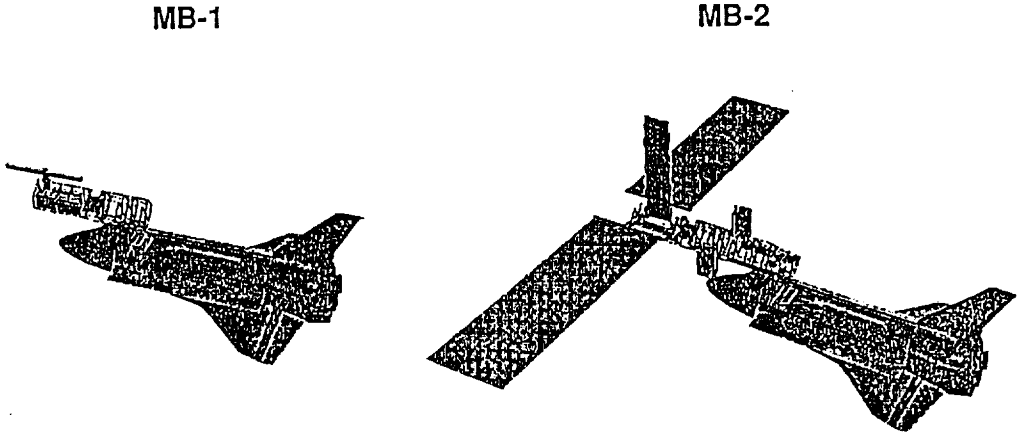


Figure 9.1 Incremental construction of space station. (Courtesy McDonnell Douglas Space Systems Co.)

9.5.1 *Manoeuvring of the SRMS with prescribed motion constraints*

Figure 9.1 illustrates the assembly of the first and second modules of the space station to be deployed and assembled. Each module is lifted by the shuttle remote manipulator system (SRMS) from the shuttle cargo bay and deployed for the first module and subsequently assembled into the partially assembled space station. In order to simulate the assembly process, first, we have studied the effect of the SRMS dynamics due to the required manoeuvring constraints. This incremental in-space construction of the space station must meet stringent geometry, weight and stiffness requirements as shown in Figure 9.2. The arm boom assemblies comprise two thin-walled graphite-epoxy circular sections called the upper arm, lower arm and end effector. These arms are connected by a shoulder joint (modelled by a universal joint), an elbow joint (modelled by a revolute joint) and a wrist joint (modelled by a spherical joint). The properties of the joints and arms are shown as follows (Hunter *et al.* [36]):

(1) Upper arm:
Young's modulus:
$E_u = 1.27 \times 10^{11}\,\mathrm{Pa}$
Shear modulus:
$G_u = 3.18 \times 10^{10}\,\mathrm{Pa}$
Length: $L_u = 6.38\,\mathrm{m}$
Cross-section area:
$A_u = 0.0022\,\mathrm{m}^2$
Moment of intertia:
$I_u = 3.16 \times 10^{-5}\,\mathrm{m}^4$
Weight: $W_u = 24.97\,\mathrm{kg}$

(2) Lower arm:
Young's modulus:
$E_l = 1.09 \times 10^{11}\,\mathrm{Pa}$
Shear modulus:
$G_l = 3.30 \times 10^{10}\,\mathrm{Pa}$
Length: $L_l = 7.06\,\mathrm{m}$
Cross-section area:
$A_l = 0.0015\,\mathrm{m}^2$
Moments of inertia:
$I_l = 2.19 \times 10^{-5}\,\mathrm{m}^4$
Weight: $W_l = 24.06\,\mathrm{kg}$

Properties of SRMS:

- Weight = 410 Kg
- Length = 15 m
- Cross Section Area = 0.0022 m^2
- Young's Module = 1.27 X 10^{11} Pa
- Shear Module = 3.18 X 10^{10} Pa
- Density = 1.2 X 10^4 Kg/m^3
- Tip Maneuvering Speed (without payload) = 0.6 m/s

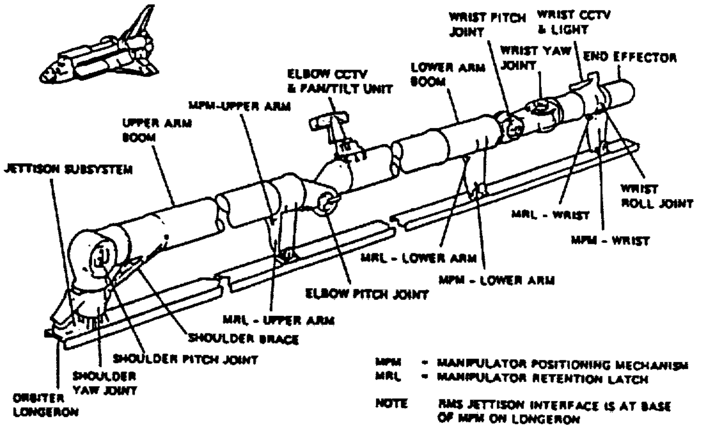


Figure 9.2 Shuttle remote manipulator system [34].

(3) End effector:
Length: $L_e = 1.82\,\mathrm{m}$
End effector weight:
$W_e = 107.14\,\mathrm{kg}$

(4) Joint weights:
Shoulder joint weight:
$W_s = 117.13\,\mathrm{kg}$
Elbow joint weight:
$W_{el} = 53.12\,\mathrm{kg}$
Wrist joint weight:
$W_w = 84.44\,\mathrm{kg}$.

The effect of the motion constraints to the orbital motion stability can be assessed by modelling (1) both the space shuttle and the SRMS modeled to be rigid, (2) the shuttle to be rigid and the SRMS as a flexible beam (discretized into 4 elements and 5 nodal points). By imposing angular velocity (a cubic type) at the tip of the SRMS (Figure 9.3), the manipulator will slew through 90° with respect to the space shuttle. Figures 9.4 and 9.5 illustrate the pitching angles of the rigid and flexible SRMS, and Figure 9.6 shows the angular velocity of the rigid and flexible SRMS. Note that the terminal velocities of the flexible case are non-zero, implying that the SRMS manoeuvring would trigger vibrations on the space shuttle modules after assembly. To overcome this difficulty, a more refined SRMS manoeuvring motion is necessary

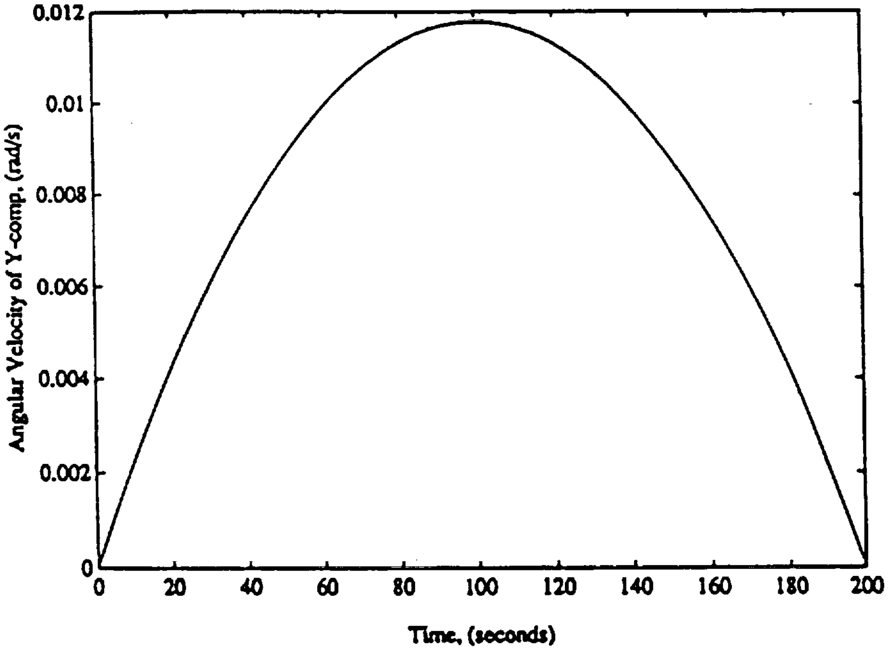


Figure 9.3 Imposed angular motions of arm and the platform.

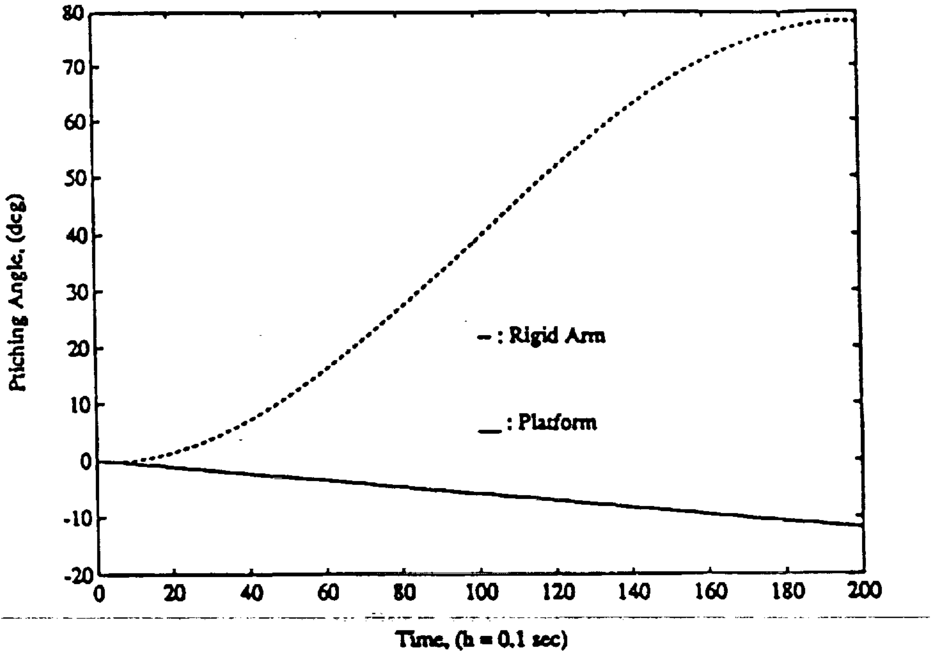


Figure 9.4 Pitching motions of arm and platform for rigid case.

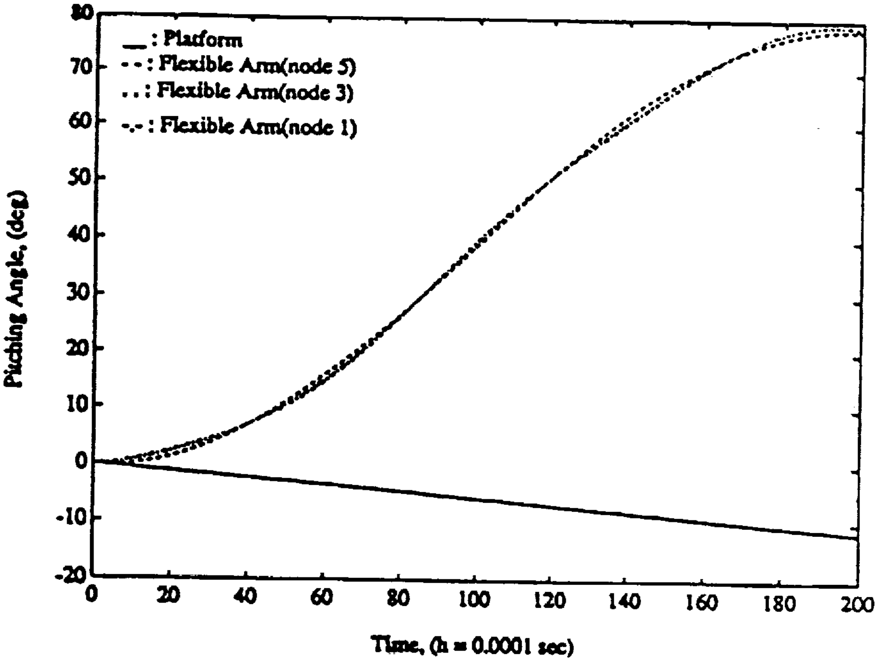


Figure 9.5 Pitching motions of arm and platform for flexible case.

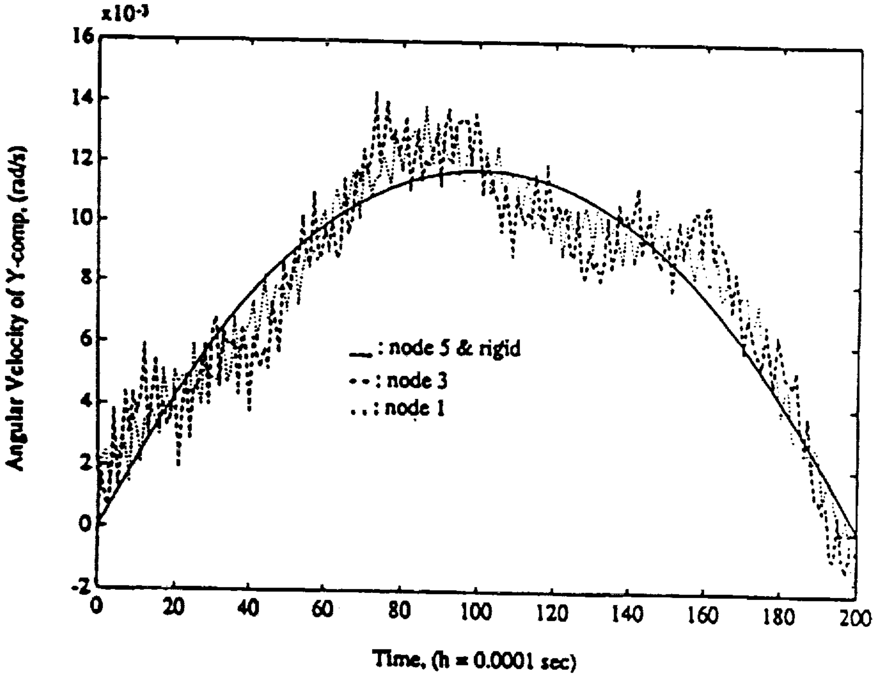


Figure 9.6 Response of yawing motion for flexible and rigid cases.

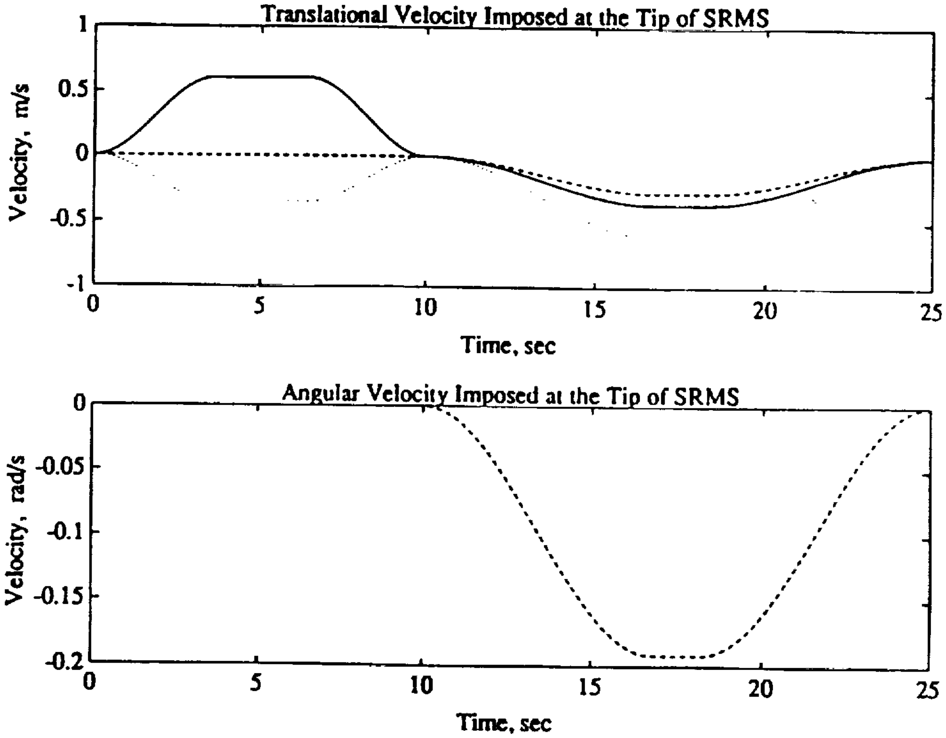


Figure 9.7 Imposed translational and angular motions at the tip of SRMS.

as described in Chiou *et al.* [37]. By adopting the refined starting and stopping conditions, the in-space construction of the space station can be divided into the following stages:

1. manoeuvring of the SRMS to the position where its end effector is ready to attach the space structure which is lying in the shuttle cargo bay;
2. contact/impact when the end effector of the SRMS collides with the space structure;
3. manoeuvring of the SRMS with the space structure to attach to another space structure which is floating in space;
4. contact/impact when the SRMS with the space structure collides with another space structure in space.

For the first stage, the motion constraints for the tip of the SRMS are given by Figure 9.7 where 25 seconds of manoeuvring time is used to place the end effector of the SRMS to the position where the space structure/payload is located. As indicated in Figures 9.8 and 9.9, the angular velocity vectors for the upper arm and lower arm of the rigid and flexible SRMS experience almost the same behaviour in terms of trends and magnitudes which prove that the present motion constraints are valid in manoeuvring the rigid and flexible SRMS. At the second stage, where the contact/impact has occurred, the end effector of the SRMS is approaching the

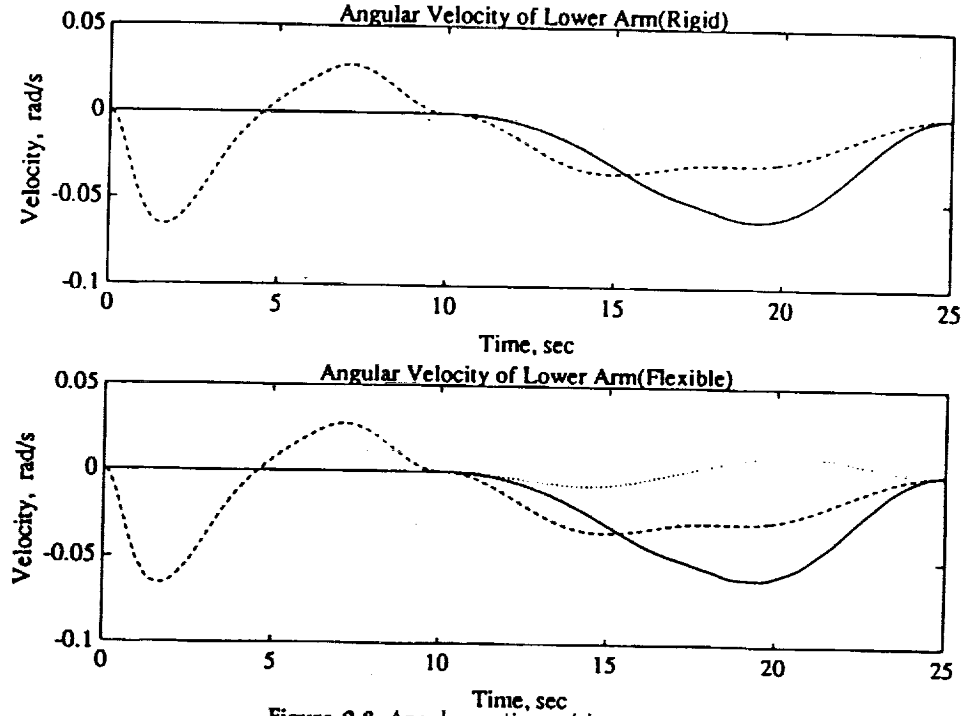


Figure 9.8 Angular motions of lower arm.

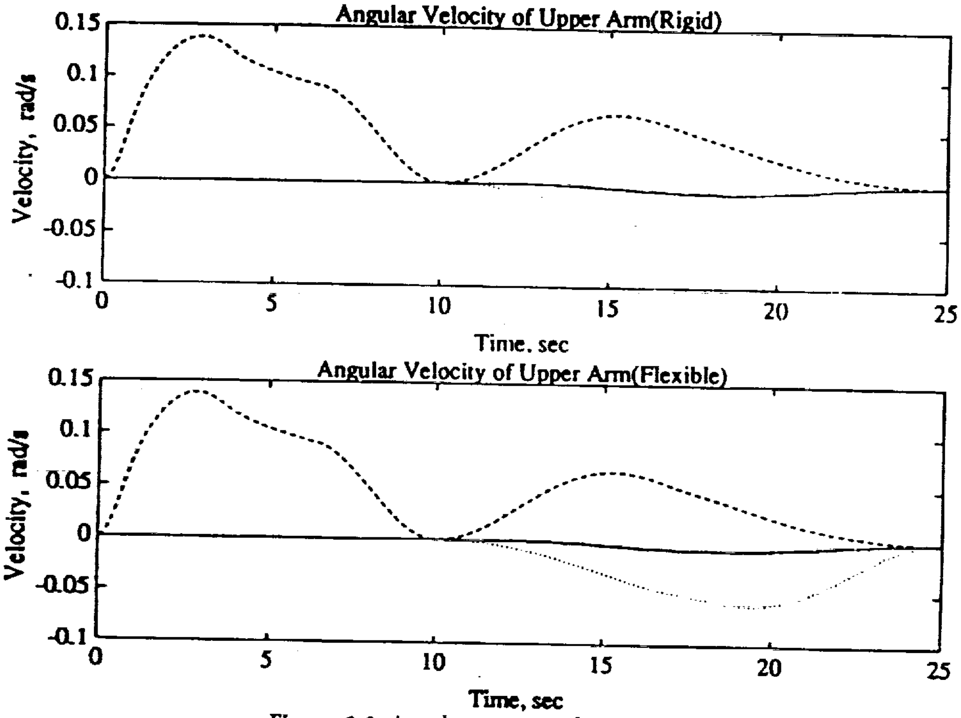


Figure 9.9 Angular motions of upper arms.

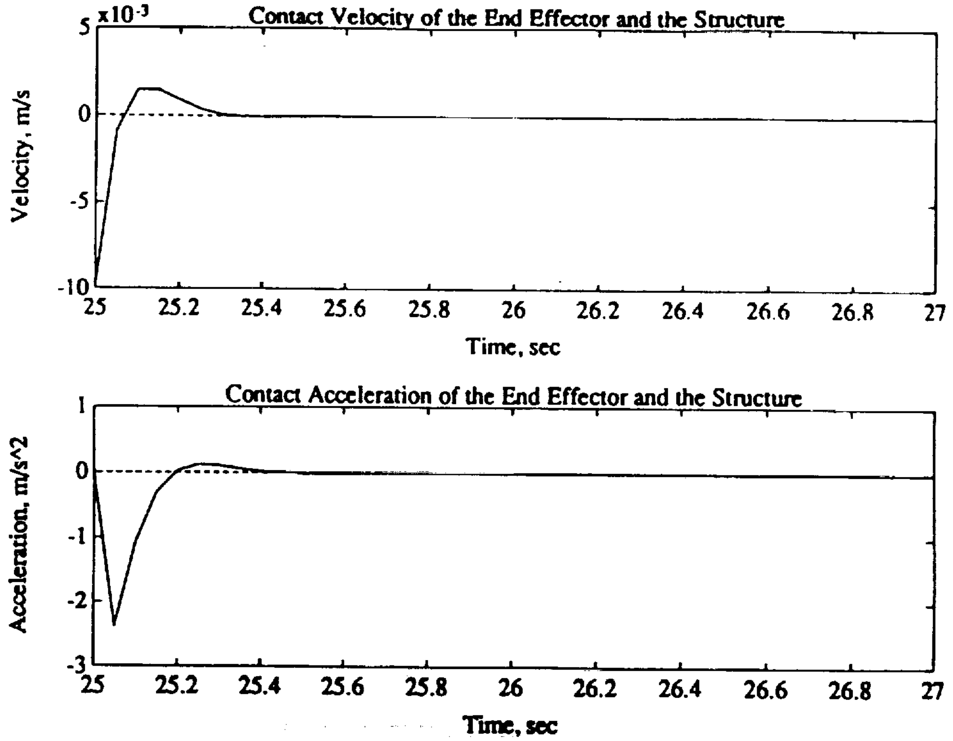
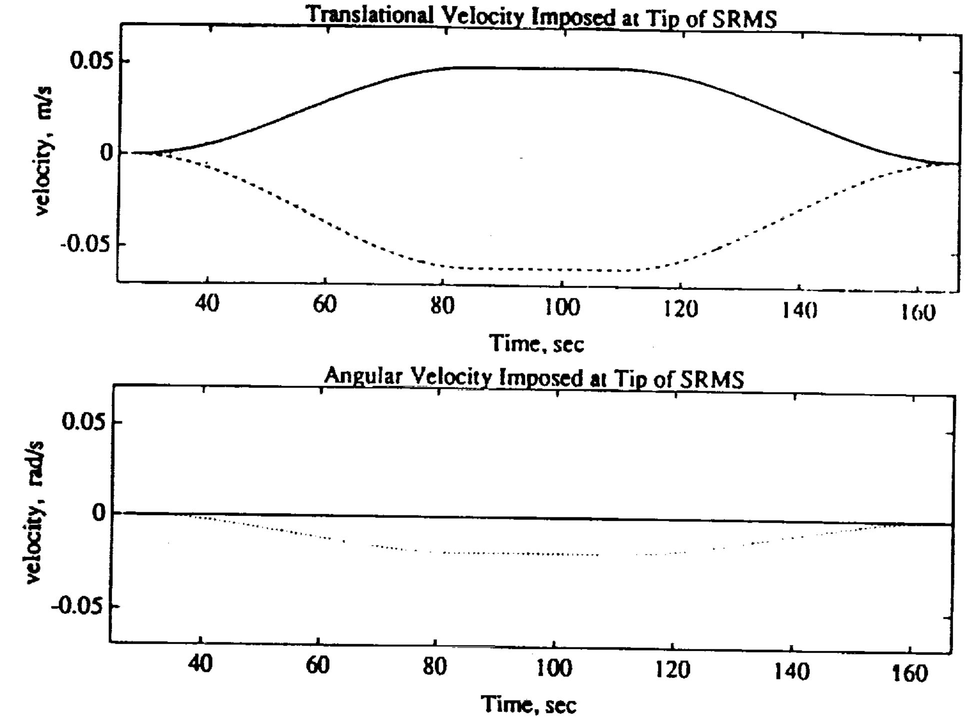


Figure 9.10 Contact velocity and acceleration of the end effector.

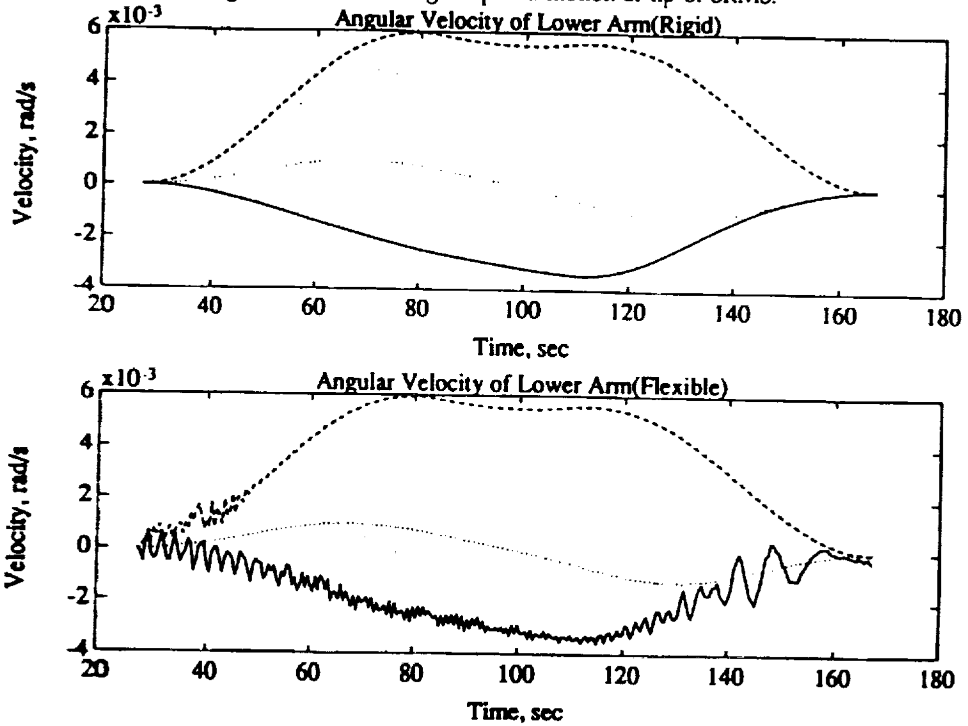
structure with velocity equal to −0.01 m/s (Figure 9.10(a)). When two bodies make contact at 25 s, the velocity of the end effector drops from −0.01 m/s to 0.0018 m/s to almost 0 m/s in less than one second of contact/impact time. From Figure 9.10(b), the contact/impact provides a peak acceleration ($-2.4\,\text{m/s}^2$) on the end effector which eventually dies down because of the large mass ratio between the end effector and the structure. At the third stage, the SRMS lifts the structure with a motion constraint that is given by Figure 9.11. The purpose of this motion constraint is to manoeuvre the structure into the position where the previously existing structure is located so that the assembly of two structures can take place via contact/impact. From Figures 9.12 and 9.13, even though the angular velocities of the flexible SRMS still maintain the trends as in the rigid SRMS case, the high vibration modes can easily be seen as the stopping conditions of the flexible SRMS are applied. Consequently, due to these vibrations, the non-zero terminal velocities have occurred, which makes the assembly of the two structures very difficult to carry out. In conclusion, since the current motion constraints cannot provide the zero terminal velocities for the flexible SRMS, the control strategy in damping out these vibrations needs to be studied in order to proceed to the final stage of the present construction process. In Figure 9.14, the contact/impact of the fourth stage has been carried out by using the rigid SRMS model that the velocity of the approaching structure is −0.01 m/s which produces of accelerations for both assembling structures during 1.8 s of contact/impact time. Note that after two seconds of contact/impact, both structures are traveling with the same velocity as indicated in Figure 9.14(a).



Translational Velocity Imposed at Tip of SRMS

Angular Velocity Imposed at Tip of SRMS

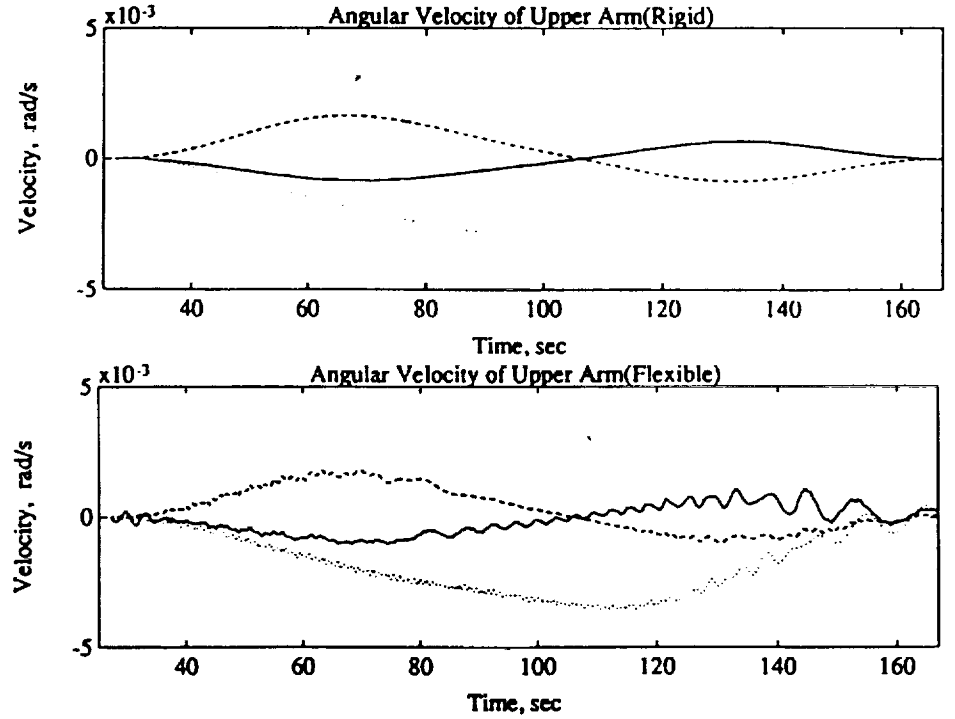
Figure 9.11 Third stage imposed motion at tip of SRMS.



Angular Velocity of Lower Arm(Rigid)

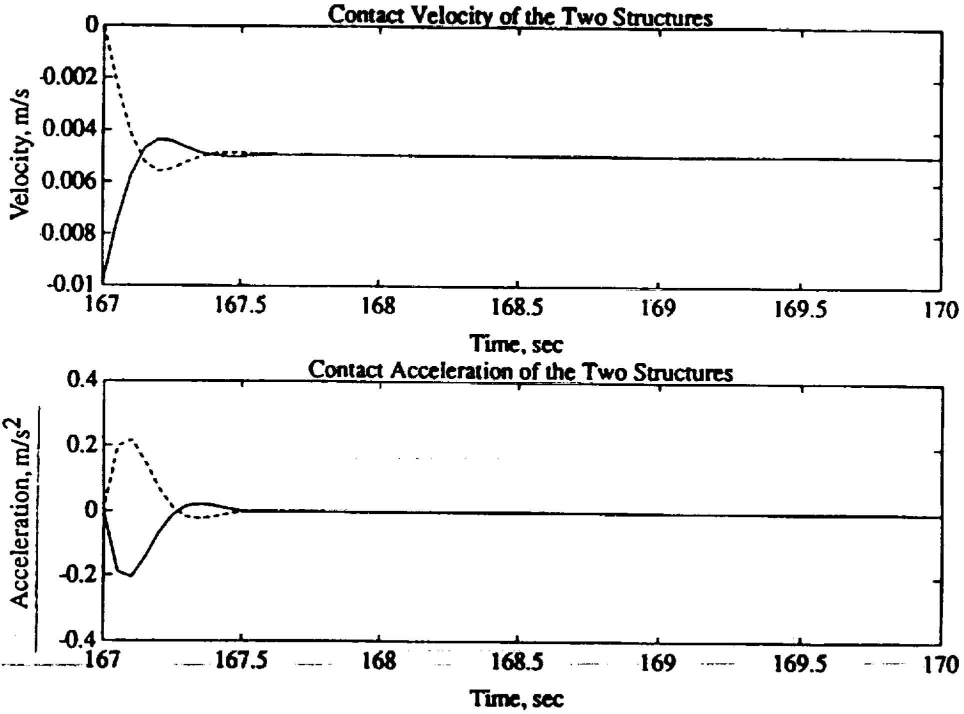
Angular Velocity of Lower Arm(Flexible)

Figure 9.12 Response of lower arm during third stage.


Angular Velocity of Upper Arm(Rigid)

Angular Velocity of Upper Arm(Flexible)

Figure 9.13 Response of upper arm during third stage.


Contact Velocity of the Two Structures

Contact Acceleration of the Two Structures

Figure 9.14 Contact velocity and acceleration during third stage.

9.5.2 *Control of earth-observing platform*

The computational efficiency achieved by the partitioned solution procedure as compared with the conventional solution method is sketched out in Figure 9.15. Assumed in the construction of the chart are the ratio of the bandwidth and the stiffness matrix to be 0.2, the number of the actuator and that of the sensor to be the same, and the ratio of the structural degrees of freedom and those of the actuator to be 0.1. Note that, given a real-time processing computer that can perform a wall-clock rate of 200 samples/second command and control, the conventional method can at most handle the real time control of a simple beam articulation, whereas the partitioned method can handle the real-time control of complex truss-beam vibrations. For nonlinear problems, the advantages of the partitioned method is more pronounced, as can be seen from the chart.

The partitioned CSI simulation procedure as derived in (9.46) has been implemented as a stand-alone package (Park *et al.* [38]). The present software implementation emphasizes the use of the widely available sequential and paralled analysis modules specially developed for the solution of structural dynamics equations. Note that the solution algorithm for both the structural system and the state estimator is the same, hence the software module, provided the right hand terms are treated as applied forces. Although the stabilized form of the controller and the filtered measurements are solved in a coupled manner, their size in general is substantially smaller, typically a fraction of the size of the structural system for large-scale problems.

Figure 9.16 illustrates a test-bed evolutionary model of an Earth-pointing satellite. Eighteen actuators and 18 sensors are applied to the system (see Figure 9.16 for their locations) for vibration control and their locations are provided in Tables 9.1 and 9.2. Figures 9.17–9.19 are representative of the responses for open-loop, direct output feedback, and dynamically compensated case does drift away initially even though the settling time is about the same as that by the direct output feedback case. However, the sensor outputs are assumed to be noise-free in these two numerical experiments. Further simulations with the present procedure should shed light on the performance of dynamically compensated feedback systems or large-scale systems as they are computationally more feasible than heretofore possible.

Tables 9.3 and 9.4 illustrate the computational overhead associated with the direct output feedback vs. the use of a dynamic compensation scheme by the output present Kalman filtering equations, compared in those tables are for two simpler tests cases, viz., a 3-d.o.f. system and a truss beam model. In the numerical experiments herein, we have relied on the Matlab software package for the synthesis of both the control law gains and the discrete Kalman filter gain matrices. Results of the full state feedback (FSFB) utilizing a direct feedback vs a dynamically compensated feedback based on the Kamlan filter (K. Filter) indicate that they become competitive as the size of the model increases. Reported in those two tables are also the effects of various implementation versions, from a nominal code (version A1) to a fully parallelized version on a shared memory machine (allied with 8 processors). The present numerical results indicate that CPU requirements for dynamically compensated CSI simulations would in general require about three times that of a typical structure-only transient analysis. It should be mentioned that even though there is a slight

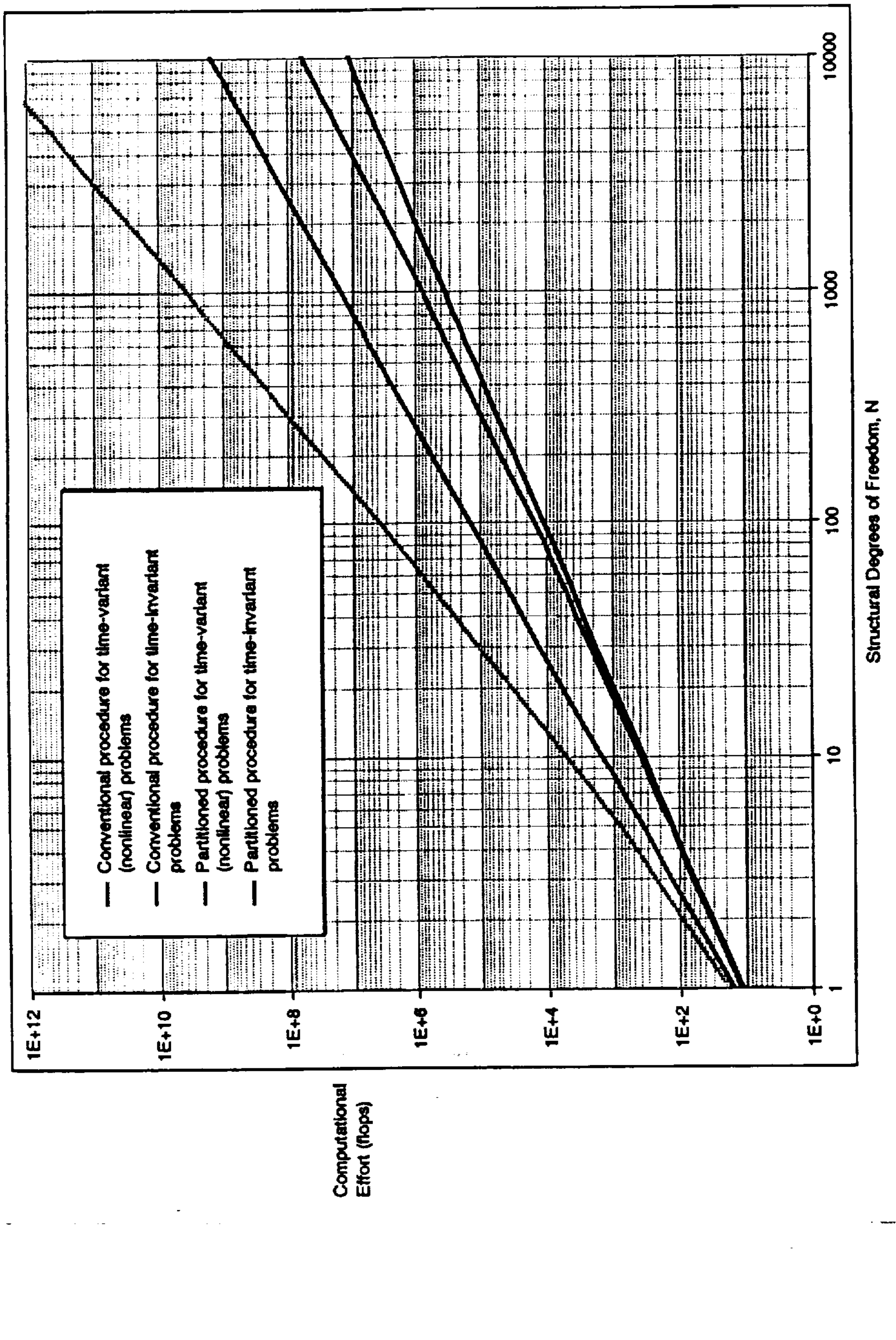


Figure 9.15 Efficiency of partitional CSI procedure.

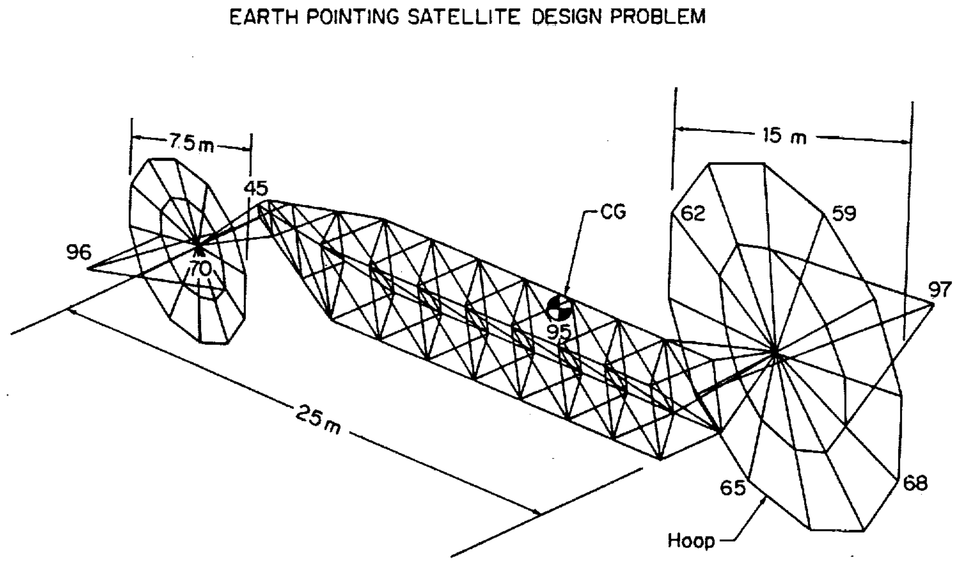


Figure 9.16 A generic Earth-pointing satellite.

Table 9.1 Actuator placement for EPS example problem.

Actuator	Node	Component
1	97	x
2	97	z
3	96	x
4	96	z
5	65	y
6	68	y
7	59	y
8	62	y
9	45	y
10	45	z
11	70	y
12	70	z
13	95	x
14	95	y
15	95	z
16	95	ϕ_x
17	95	ϕ_y
18	95	ϕ_z

increase of total CPU units from the compiler optimized sequential run to the parallel case, the actual run-clock time on the parallel machine is about one-eighth of the sequential case.

To conclude, it is seen that the use of the present second-order discrete Kalman

Table 9.2 Sensor placement for EPS example problem.

Sensor	Type	Node	Component
1	Rate	97	x
2	Rate	97	z
3	Rate	96	x
4	Rate	96	z
5	Rate	65	y
6	Rate	68	y
7	Rate	59	y
8	Rate	62	y
9	Rate	45	y
10	Rate	45	z
11	Rate	70	y
12	Rate	70	z
13	Position	95	x
14	Position	95	y
15	Position	95	z
16	Position	95	ϕ_z
17	Position	95	ϕ_y
18	Position	95	ϕ_z

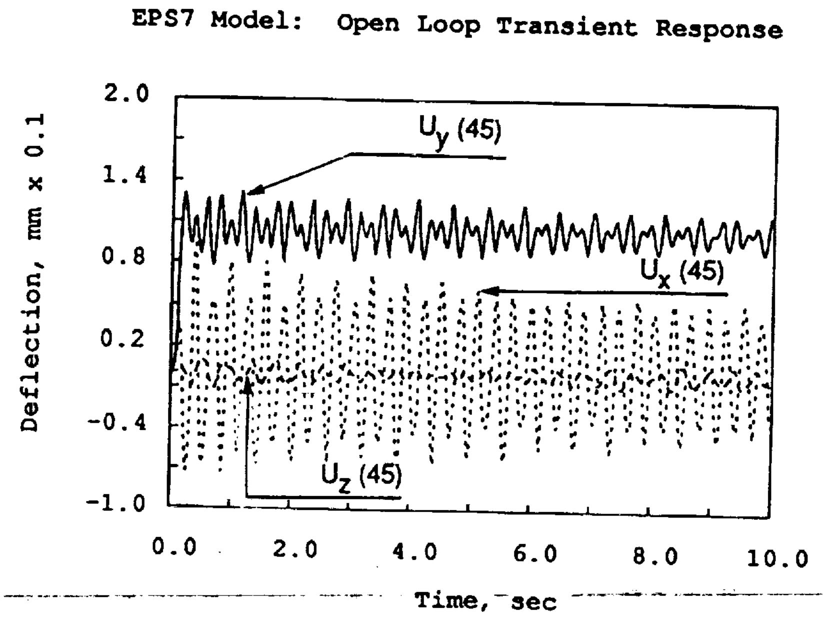


Figure 9.17 Open loop transient response.

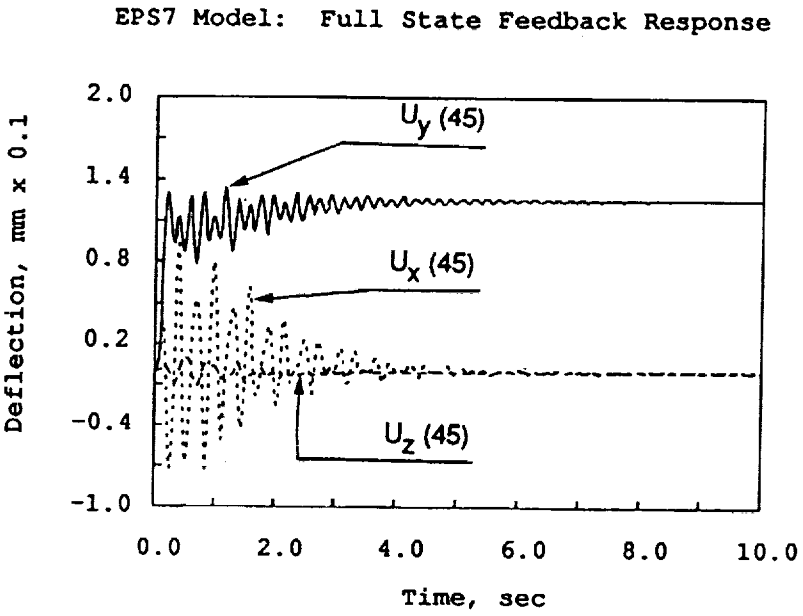


Figure 9.18 Full state feedback response.

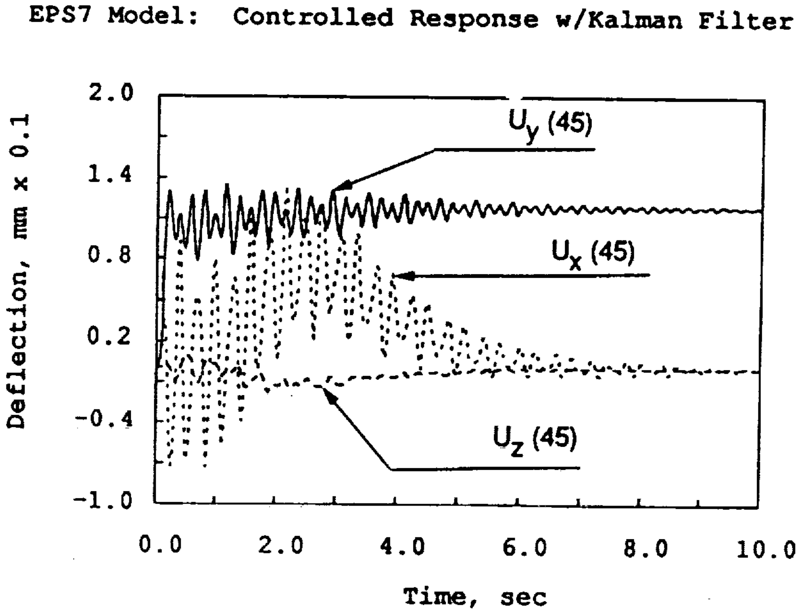


Figure 9.19 Dynamically compensated response via Kalman filter.

Table 9.3 CPU results for versions of ACSIS.

Model	Problem type	(A1) Nominal code	(A2) Compiler optimized	(A3) Parallel observer
3DOF	Transient	6.6	2.1	2.1
Spring	FSFB	8.0	3.3	3.3
	K. Filter	12.3	3.5	3.3
54 DOF	Transient	78.2	5.7	5.6
Truss	FSFB	97.1	9.4	10.2
	K. Filter	170.7	13.0	10.7
582 DOF	Transient	3506	98.6	100.3
EPS7	FSFB	7040	190.2	294.5
	K. Filter	n/a	284.2	312.5

Table 9.4 CPU results for ACSIS with EBE computations.

Model	Problem type	(A4) E-B-E computation	(A5) Parallel E-B-E	(A6) Parallel Obs. & EBE
3 DOF	Transient	3.8	3.3	3.3
Spring	FSFB	4.9	4.4	4.9
	K. Filter	6.6	5.6	5.0
54 DOF	Transient	31.7	13.0	13.0
Truss	FSFB	35.5	16.9	35.6
	K. Filter	62.6	27.3	36.2
582 DOF	Transient	391.7	153.9	n/a
EPS7	FSFB	485.9	245.9	n/a
	K. Filter	n/a	n/a	n/a

filtering equations for constructing dynamically compensated control laws add computational overhead, but is only the equivalent of open-loop transient analysis of symmetric sparse systems of order N instead of $2N \times 2N$ dense systems.

9.6 CLOSING REMARKS

The present survey have focused mostly on the research activities on the computational methods for multibody dynamics, control-structure interactions, and coupled thermal–structural transients undertaken by the researchers at the Center for Space Structures and Controls, University of Colorado. No attempt has thus been made to include many important advances made by researchers around the world. A notable omission in this survey is computational methods for parallel computations, which is intensely pursued by many researchers. We hope to report on the progress on parallel computational methods at a later occasion.

ACKNOWLEDGEMENTS

This survey consists of work performed by the author's colleagues, in alphabetical order, K. F. Alvin, W. K. Belvin, J. C. Chiou, J. D. Downer, C. Farhat and C. A. Felippa. Without having had the privilege of working with them over the past six years, it would not have been possible for the author to put together the present survey. He is deeply indebted to them for this privilege. However, any misconception or imbalance in the coverage of the materials solely rests on the author and he hopes they will forgive him for not presenting their work as they conceive to be the case if any misrepresentation has been made. This survey was prepared while the author was visiting the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, during Spring 1992. He thanks Prof. E. F. Crawley and his staff for providing a nurturing environment for his sabbatical stay at the Space Engineering Research Center, a NASA-sponsored space engineering center at MIT.

BIBLIOGRAPHY

[1] Park, K.C. (1988) *Transient Analysis Methods in Computational Dynamics*, in D.L. Dwoyer, M.Y. Hussaini and R.G. Voigt (eds.), *Finite Elements: Theory and Applications*, Springer-Verlag, pp. 240–267.

[2] Park, K.C. and Felippa, C.A. (1983) Partitioned analysis of coupled systems, in T. Belytschko and T.J.R. Hughes (eds.), *Computational Methods for Transient Analysis*, Elsevier, pp. 157–219.

[3] Felippa, C.A. and Park, K.C. (1980) Staggered transient analysis procedures for coupled-field mechanical systems: formulation, *Comput. Meth. Appl. Mech. Engg*, **24**, 61–111.

[4] Park, K.C. (1977) Practical aspects of numerical time integration, *Comput. Struct.*, **7**, 343–353.

[5] Hughes, T.J.R. and Belytschko, T. (1985) A précis of developments in computational methods for transient analysis, *J. Appl. Mech.*, **50**, 1033–1041.

[6] Park, K.C. and Chiou, J.C. (1988) Stabilization of computational procedures for constrained dynamical systems, *J. Guidance, Control and Dynamics*, **11**, 365–370.

[7] Park, K.C., Chiou, J.C. and Downer, J.D. (1990) Explicit–implicit staggered procedure for multibody dynamics analysis, *J. Guidance, Control, and Dynamics*, **13**, 562–570.

[8] Park, K.C., Downer, J.D., Chiou, J.C. and Farhat, C. (1991) A modular multibody analysis capability for high-precision, active control and real-time applications, *Int. J. Num. Meth. Engg*, **32**, 1767–1798.

[9] Wehage, R.A. and Haug, E.J. (1982) Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, *ASME J. Mech. Design*, **104**, 247–255.

[10] Chiou, J.C. (1990) *Constraint Treatment Techniques and Parallel Algorithms for Multibody Dynamic Analysis*, Ph.D. Thesis, University of Colorado.

[11] Chiou, J.C., Park, K.C. and Farhat, C. (1991) A natural partitioning scheme for parallel simulation of multibody systems, *1991 AIAA Structures, Dynamics and Materials Conference*, Paper No. AIAA-91-1111, Baltimore, Maryland, 8–10 April 1991.

[12] Baumgarte, J.W. (1972) Stabilization of constraints and integrals of motion in dynamical systems, *Comput. Meth. Appl. Mech. Engg*, **1**, 1–16.

[13] Baumgarte, J.W. (1983) A new method of stabilization for holonomic constraints, *J. Appl. Mech.*, **50**, 869–870.

[14] Park, K.C. and Chiou, J.C. (1992) A Discrete Momentum-Conserving Explicit Algorithm for Multibody Dynamic Analysis, to appear in *Int. J. Num. Meth. Engg.*

[15] Simo, J.C. and Wong, K.K. (1991) Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum, Int. J. Num. Meth. Engg, **31**(1), 19–52.

[16] Downer, J.D. (1990) *A Computational Procedure for the Dynamics of Flexible Beams within Multibody Systems*, Ph.D. Thesis, University of Colorado.

[17] Downer, J.D., Park, K.C. and Chiou, J.C. (1991) A Computational Procedure for Multibody Systems Including Flexible Beam Dynamics, to appear in *Comput. Meth. Appl. Mech. Engg, Proc. the 1990 AIAA Dynamics Specialist Conference*, Paper No. AIAA-90-1237, Long Beach, California, 5–6 April, 1990.

[18] Downer, J.D. and Park, K.C. (1991) Dynamics of Spacecraft with Deploying Flexible Appendages, *Comput. Meth. Appl. Mech. Engg*, 1991; *Proc. the 1992 AIAA Dynamics Specialist Conference*, Paper No. AIAA-92-2087, Dallas, Texas, 16–17, April 1992.

[19] Haug, E.J. and Deyo, R.C. (eds.) (1991) *Real-Time Integration Methods for Mechanical Systems Simulation*, NATO, ASI Series, Springer-Verlag, Berlin.

[20] Belvin, W.K. (1989) *Simulation and Interdisciplinary Design Methodology for Control-Structure Interaction Systems*, Ph.D. Thesis, Center for Space Structures and Controls, University of Colorado, Report No. CU-CSSC-89-10, July.

[21] Park, K.C. and Belvin, W.K. (1991) A partitioned solution procedure for control-structure interaction simulations, *J. Guidance, Control and Dynamics*, **14**(1), January–February, 59–67.

[22] Park, K.C., Felippa, C.A. and DeRuntz, J.A. (1977) Stabilization of staggered solution procedures for fluid-structure interaction analysis, in Belytschko, T. and Geers, T.L. (eds.) *Comput. Meth. Fluid-Structure Interaction Problems*, ASME, AMD Vol. 26, New York, N.Y., pp. 95–124.

[23] Park, K.C. (1980) Partitioned analysis procedures for coupled-field problems: stability analysis, *J. Appl. Mech.*, **47**, 370–378.

[24] Zienkiewicz, O.C., Paul, D.K. and Chan, A.H.C. (1988) Unconditionally stable staggered solution procedure for soil-pore fluid interaction problems, *Int. J. Num. Meth. Engg*, **26**, 1039–1055.

[25] Kwarkemaak, H. and Sivan, R. (1974) *Linear Optimal Control Systems*, Wiley-Interscience, New York.

[26] Kalman, R.E. and Bucy, R.S. (1961) New results in linear filtering and prediction theory, *ASME J. Basic Engg*, **83**, 95–108.

[27] Luenberger, D.G. (1974) Observing the state of a linear system, *IEEE Trans. Military Electronics*, **8**, 74–80.

[28] Belvin, W.K. and Park, K.C. (1989) On the state estimation of structures with second order observers, *Proc. 30th Structures, Structural Dynamics and Materials Conference*, AIAA Paper No. 89-1241, April 3–5.

[29] Gantmacher, F.R. (1959) *The Theory of Matrices*, 2, Chelsea, New York, pp. 190–196.

[30] Wilson, E.L. and Nickell, R.E. (1966) Application of the finite element method to heat conduction analysis, *Nucl. Eng. Des.*, **4**, 276–286.

[31] Nickell, R.E. and Sackman, J.L. (1968) The extended Ritz method applied to transient coupled thermoelastic boundary-value problems, *J. Appl. Mech.*, **35**, 255–266.

[32] Oden, J.T. (1969) Finite element analysis of nonlinear problems in the dynamical theory of coupled thermoelasticity, *Nucl. Eng. Des.*, **10**, 465–475.

[33] Oden, J.T. and Armstrong, W.H. (1971) Analysis of nonlinear dynamic coupled thermoviscoelasticity problems by the finite element method, *Comput. Struct.*, **1**, 603–621.

[34] Farhat, C., Park, K.C. and Dubois-Pelerin, Y.D. (1991) An unconditionally stable

staggered algorithm for transient finite-element analysis of coupled thermoelastic problems, *Comput. Meth. Appl. Mech. Engg*, **85**, 349–365.

[35] Bellman, R. (1970) *Introduction to Matrix Analysis* (2nd edn), McGraw-Hill, New York, pp. 249–262.

[36] Hunter, J.A., Ussher, T.H. and Gossain, D.M. (1982) Structural dynamic design consideration of the shuttle remote manipulator system, *Proc. the 1982 AIAA Structures, Structural Dynamics and Materials Conf.*, Paper No. AIAA-82-0762, pp. 499–505.

[37] Chiou, J.C., Downer, J.D., Natori, M.C. and Park, K.C. (1992) Interaction dynamics of an orbiter and a flexible space structure undergoing incremental in-space construction, to be presented at the 32nd SDM Conf., Dallas, TX, April 1992.

[38] Park, K.C., Belvin, W.K. and Alvin, K.F. (1992) Second-order discrete Kalman filtering equations for control-structure interaction simulations, to appear in *Journal of Guidance, Control, and Dynamics*.

K.C. Park
Department of Aerospace Engineering Sciences and Center for Space Structures and Controls
University of Colorado at Boulder
Campus Box 429
Boulder
CO 80309-0429
USA

Discrete Kalman Filtering Equations of Second-Order Form for Control-Structure Interaction Simulations+

K. C. Park[1] and K. F. Alvin[2]
Center for Space Structures and Controls
University of Colorado, Campus Box 429
Boulder, Colorado 80309

and

W. Keith Belvin[3]
Spacecraft Dynamics Branch
NASA Langley Research Center
Hampton, Virginia 23665

Abstract

A second-order form of discrete Kalman filtering equations is proposed as a candidate state estimator for efficient simulations of control-structure interactions in coupled physical coordinate configurations as opposed to decoupled modal coordinates. The resulting matrix equation of the present state estimator consists of the same symmetric, sparse $N \times N$ coupled matrices of the governing structural dynamics equations as opposed to unsymmetric $2N \times 2N$ state space-based estimators. Thus, in addition to substantial computational efficiency improvement, the present estimator can be applied to control-structure design optimization for which the physical coordinates associated with the mass, damping and stiffness matrices of the structure are needed instead of modal coordinates.

+ An earlier version of the present paper without numerical experiments was presented at the AIAA Guidance and Control Conference, Portland, Ore., 20-22 August 1990, Paper No. AIAA 90-3387.

[1] Professor of Aerospace Engineering, University of Colorado. Associate Fellow of AIAA.

[2] Graduate Research Assistant

[3] Structural Dynamics Division, NASA Langley Research Center. Senior Member AIAA.

Introduction

Current practice in the design, modeling and analysis of flexible large space structures is by and large based on the finite element method and the associated software. The resulting discrete equations of motion for structures, both in terms of physical coordinates and of modal coordinates, are expressed in a second-order form. As a result, the structural engineering community has been investing a considerable amount of research and development resources to develop computer-oriented discrete modeling tools, analysis methods and interface capabilities with design synthesis procedures; all of these exploiting the characteristics of second-order models. Recent work in the area of structural dynamics simulation and massively-parallel processing also rely on the second-order equation forms.

On the other hand, modern linear control theory has its roots firmly in a first-order form of the governing differential equations, e.g., Kwakernaak and Sivan[1]. Thus, several investigators have addressed the issues of interfacing second-order structural systems and control theory based on the first-order form[2-7]. As a result of these studies, it has become straightforward for one to synthesize direct state feedback based control laws within the framework of a first-order control theory and then to recast the resulting control laws in terms of the second-order structural systems.

Unfortunately, controllers based on a first-order state estimator are difficult to express in a pure second-order form because the first-order estimator implicitly incorporates an additional filter equation[7]. However a recent work by Juang and Maghami[8] has enabled the first-order filter gain matrices to be synthesized using only second-order equations. To complement the second-order gain synthesis, the objective of the present paper is to develop a second-order based simulation procedure for first-order estimators. The particular class of first-order dynamic compensation chosen for study are the Kalman Filter based state estimators as applied to second-order structural systems. The proposed procedure permits simulation of first-order estimators with nearly the same solution procedure used for treating the structural dynamics equation. Hence, the reduced size of system matrices and the computational techniques that are tailored to sparse second-order structural systems may be employed. As will be shown, the proposed procedure hinges on discrete time integration formulas to effectively reduce the continuous time Kalman Filter to a set of second-order difference equations.

The *primary goal* of the proposed procedure is the incorporation of this general form of state estimation as a *simulation tool* in partitioned control-structure interaction (CSI) analyses. It is expected that Kalman Filters for real-time control of linear time-invariant systems would be implemented in the most efficient form available, typically a real mode-decoupled state space realization. For analytical studies of CSI systems, however, where the objective is frequently simultaneous optimization of controls and structures as in Belvin[9], the use of such a modal form must be weighed against the preprocessing tasks required to generate the model. In these cases, much more flexible controllers expressed in terms of the physical coordinates instead of the modal coordinates are sought, ones which can readily adapt to iterative changes in the structural parameters. One such control law synthesis has been proposed and demonstrated to be effective for CSI optimization[7]. These studies did not account, however, for dynamic compensation when full state feedback control was utilized. With the discrete Kalman Filter proposed herein, a general form of dynamic compensation can be integrated into CSI simulation and optimization which does not impose limits on the designs of the feedback gains or the filter gains.

The paper first reviews of the conventional first-order representation of the continuous second-order structural equations of motion, in which the state variables are defined as the displacements variables x of the second-order structural model and the velocities $\dot{x}$. An examination of the corresponding first-order Kalman filtering equations indicates that, due to the difference in the derivative of the estimated displacement ($\frac{d}{dt}\hat{x}$) and the

estimated velocity ($\dot{\hat{x}}$), transformation of the first-order estimator into an equivalent second-order estimator requires the time derivative of measurement data, a process not recommended for practical implementation.

Next, a transformation via a generalized momentum is introduced to recast the structural equations of motion in a general first-order setting. It is shown that discrete time numerical integration followed by reduction of the resulting difference equations circumvents the need for the time derivative of measurements to solve Kalman filtering equations in a second-order framework. Hence, the Kalman filter equations can be solved using a second-order solution software package.

Subsequently, computer implementation aspects of the present second-order estimator are presented. Several computational paths are discussed in the context of discrete and continuous time simulation. For continuous time control simulation, an equation augmentation is introduced to exploit the symmetry and sparsity of the attendant matrices by maintaining state dependent control and observer terms on the right-hand-side (RHS) of the filter equations. In addition, the computational efficiency of the present second-order filter as compared to the first-order form is presented.

Continuous Formulation of State Estimators for Structural Systems

Linear, second-order discrete structural models can be expressed as

$$M\ddot{x} + D\dot{x} + Kx = Bu + Gw\,, \qquad x(0) = x_0\,, \quad \dot{x}(0) = \dot{x}_0 \tag{1}$$

$$u = -Z_1 x - Z_2 \dot{x}$$

with the associated measurements

$$z = H_1 x + H_2 \dot{x} + \nu \tag{2}$$

where M, D, K are the mass damping and stiffness matrices of size $(N \times N)$; x is the structural displacement vector, $(N \times 1)$; u is the active control force $(m \times 1)$; B is a constant force distribution matrix $(N \times m)$; z is a set of measurements $(r \times 1)$; H_1 and H_2 are the measurement distribution matrices $(r \times N)$; Z_1 and Z_2 are the control feedback gain matrices $(m \times N)$; w and ν are zero-mean, white Gaussian processes with their respective covariances Q and R; and the superscript dot designates time differentiation. In the present study, we will restrict ourselves to the case wherein Q and R are uncorrelated with each other and the initial conditions x_0 and $\dot{x}_0$ are also themselves jointly Gaussian with known means and covariances.

The conventional representation of (1) in a first-order form is facilitated by

$$\begin{cases} x_1 = x \\ x_2 = \dot{x} = \dot{x}_1 \\ M\dot{x}_2 = M\ddot{x} = Bu + Gw - Dx_2 - Kx_1 \end{cases} \tag{3}$$

which, when cast in a first-order form, can be expressed as

$$\begin{cases} E\dot{q} = Fq + \bar{B}u + \bar{G}w, \quad q = (\,x_1 \quad x_2\,)^T \\ z = Hq + \nu \end{cases} \tag{4}$$

where

$$E = \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix}, \quad F = \begin{bmatrix} 0 & I \\ -K & -D \end{bmatrix},$$

$$\bar{B} = \begin{Bmatrix} 0 \\ B \end{Bmatrix}, \quad \bar{G} = \begin{Bmatrix} 0 \\ G \end{Bmatrix} \tag{5}$$

It is well-known that the Kalman filtering equations[10,11] for (4) can be shown to be (see, e.g., Arnold and Laub[3]):

$$E\dot{\hat{q}} = F\hat{q} + \bar{B}u + EPH^T R^{-1}\bar{z} \tag{6}$$

where

$$\bar{z} = z - H\hat{q}\,, \quad P = \begin{bmatrix} U & S^T \\ S & L \end{bmatrix}, \quad \hat{q} = \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix} = \begin{Bmatrix} \hat{x} \\ \hat{\dot{x}} \end{Bmatrix} \tag{7}$$

in which U and L are positive definite matrices, $\hat{q}$ is the state estimation vector, and the matrix P is determined by the Riccati equation[1,3]

$$E\dot{P}E^T = FPE^T + EPF^T - EPH^T R^{-1} HPE^T + \bar{G}Q\bar{G}^T \tag{8}$$

The inherent difficulty of reducing the first-order Kalman filtering equations given by (6) to second order form can be appreciated if one attempts to write (6) in a form introduced in (3):

$$\begin{cases} a) & \hat{x}_1 = \hat{x} \\ b) & \hat{x}_2 = \hat{\dot{x}} = \dot{\hat{x}}_1 - L_1\bar{z} \\ c) & M\dot{\hat{x}}_2 = -D\hat{x}_2 - K\hat{x}_1 + B\hat{u} + ML_2\bar{z} \end{cases} \tag{9}$$

where

$$L_1 = (H_1U + H_2S)^T R^{-1}, \qquad L_2 = (H_1S^T + H_2L)^T R^{-1}$$

Note from (9b) that $\hat{x}_2 \neq \dot{\hat{x}}_1$. In other words, the time derivative of the estimated displacement ($\dot{\hat{x}}$) is not the same as the estimated velocity ($\hat{\dot{x}}$); hence, $\hat{x}_1$ and $\hat{x}_2$ must be treated as two independent variables, an important observation somehow overlooked in Hashemipour and Laub[12].

Of course, although not practical, one can eliminate $\hat{x}_2$ from (9). Assuming $\dot{\hat{x}}_1$ and $\hat{x}_2$ are differentiable, differentiate (9b) and multiply both sides by M to obtain

$$M\ddot{\hat{x}}_1 = M\dot{\hat{x}}_2 + ML_1\dot{\bar{z}} \tag{10}$$

Substituting $M\dot{\hat{x}}_2$ from (9c) and $\hat{x}_2$ from (9b) in (10) yields

$$M\ddot{\hat{x}}_1 = -D(\dot{\hat{x}}_1 - L_1\bar{z}) - K\hat{x}_1 + Bu + ML_2\bar{z} + ML_1\dot{\bar{z}} \tag{11}$$

which, upon rearrangements, becomes

$$M\ddot{\hat{x}}_1 + D\dot{\hat{x}}_1 + K\hat{x}_1 = Bu + ML_2\bar{z} + ML_1\dot{\bar{z}} + DL_1\bar{z} \tag{12}$$

There are two difficulties with the above second-order estimator. First, the numerical solution of (12) involves the computation of $\ddot{\hat{x}}_1$ when rate measurements are made. The accuracy of this computation is in general very susceptible to errors caused in numerical differentiation of $\dot{\hat{x}}_1$. Second, and most important, the numerical evaluation of $\dot{\bar{z}}$ that is required in (12) assumes that the derivative of measurement information is available which should be avoided in practice. We now present a computational procedure that circumvents the need for computing measurement derivatives and that enables one to construct estimators based on the second-order model form.

Second-Order Transformation of Continuous Kalman Filtering Equations

This section presents a transformation of the continuous time first-order Kalman filter to a discrete time set of second-order difference equations for digital implementation. The procedure avoids the need for measurement derivative information. In addition, the sparsity and symmetry of the original mass, damping and stiffness matrices can be maintained. Prior to describing the numerical integration procedure, a transformation based on generalized momenta is presented which is later used to improve computational efficiency of the equation solution.

Generalized Momenta

Instead of the conventional transformation (3) of the second-order structural system (1) into a first-order form, let us consider the following generalized momenta (see, e.g., Jensen[13] and Felippa and Park[14]):

$$\left\{\begin{array}{ll} a) & x_1 = x \\ b) & x_2 = AM\dot{x}_1 + Cx_1 \end{array}\right. \tag{13}$$

where A and C are constant matrices to be chosen. Note that AM should be nonsingular in order to obtain an equivalent form of (1). Time differentiation of (13b) yields

$$\dot{x}_2 = AM\ddot{x}_1 + C\dot{x}_1 \tag{14}$$

Substituting (1) via (13a) into (14), one obtains

$$\dot{x}_2 = A(Bu + Gw) - (AD - C)\dot{x}_1 - AKx_1 \tag{15}$$

Finally, pairing of (13b) and (15) gives the following first-order form:

$$\begin{bmatrix} AM & 0 \\ AD - C & I \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} C & -I \\ AK & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} 0 \\ A(Bu + Gw) \end{bmatrix} \tag{16}$$

The associated Kalman filtering equation can be shown to be of the following form:

$$\begin{bmatrix} AM & 0 \\ AD - C & I \end{bmatrix} \begin{Bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{Bmatrix} + \begin{bmatrix} C & -I \\ AK & 0 \end{bmatrix} \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ ABu \end{Bmatrix} + \begin{bmatrix} AM & 0 \\ AD - C & I \end{bmatrix} \begin{bmatrix} \bar{L}_1 \\ \bar{L}_2 \end{bmatrix} \bar{z} \tag{17}$$

where

$$\bar{L}_1 = (\bar{H}_1 U + \bar{H}_2 S)^T R^{-1}, \qquad \bar{L}_2 = (\bar{H}_1 S^T + \bar{H}_2 L)^T R^{-1}$$

and $\bar{H}_1$ and $\bar{H}_2$ correspond to a modified form of measurements expressed as

$$z = H_1 x + H_2 \dot{x} = \bar{H}_1 x_1 + \bar{H}_2 x_2 \tag{18}$$

where

$$\bar{H}_1 = H_1 - H_2 M^{-1} A^{-1} C, \qquad \bar{H}_2 = H_2 M^{-1} A^{-1}$$

Clearly, as in the conventional first-order form (9), $\hat{x}_1$ and $\hat{x}_2$ in (17) are now two independent variables. Specifically, the case of $A = M^{-1}$ and $C = 0$ corresponds to (3) with $x_2 = \dot{x}_1$. However, as we shall see below, the Kalman filtering equations based on the generalized momenta (13) offer several computational advantages over (3).

Numerical Integration

At this juncture it is noted that in the previous section one first performs the elimination of $\hat{x}_1$ in order to obtain a second-order equation, then performs the numerical solution of the resulting equation. This approach has the disadvantage of having to deal with the time derivative of measurement data. To avoid this, we will first integrate numerically the associated Kalman filtering equation (17).

The direct time integration formula we propose to employ is a mid-point version of the trapezoidal rule:

$$\begin{cases} a) & \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix}^{n+1/2} = \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix}^{n} + \delta \begin{Bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{Bmatrix}^{n+1/2} \\ b) & \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix}^{n+1} = 2 \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix}^{n+1/2} - \begin{Bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{Bmatrix}^{n} \end{cases} \tag{19}$$

where the superscript n denotes the discrete time interval $t^n = nh$, h is the time increment and $\delta = h/2$.

It should be noted that we have chosen the trapezoidal rule due to its unconditional stability and high accuracy while it does not introduce any numerical damping (see, for example, Dahlquist[15] and Park[16]). Contamination of damping from numerical dissipation can not only adversely affect the solution accuracy but lead to misinterpretation of the simulation results.

Time discretization of (17) by (19a) at the $n + 1/2$ time step yields

$$\begin{aligned} &\begin{bmatrix} AM & 0 \\ AD - C & I \end{bmatrix} \begin{Bmatrix} \hat{x}_1^{n+1/2} - \hat{x}_1^n \\ \hat{x}_2^{n+1/2} - \hat{x}_2^n \end{Bmatrix} + \delta \begin{bmatrix} C & -I \\ AK & 0 \end{bmatrix} \begin{Bmatrix} \hat{x}_1^{n+1/2} \\ \hat{x}_2^{n+1/2} \end{Bmatrix} \\ &\quad = \delta \begin{bmatrix} AM & 0 \\ AD - C & I \end{bmatrix} \begin{bmatrix} \bar{L}_1 \\ \bar{L}_2 \end{bmatrix} \bar{z}^{n+1/2} + \delta \begin{Bmatrix} 0 \\ ABu^{n+1/2} \end{Bmatrix} \end{aligned} \tag{20}$$

The above difference equations require the solution of matrix equations of $2N$ variables, namely, in terms of the two variables $\hat{x}_2^{n+1/2}$ and $\hat{x}_1^{n+1/2}$, each with a size of N. To reduce the above coupled equations of order $2N$ into the corresponding ones of order N, we proceed in the following way by exploiting the nature of parametric matrices of A and C as introduced in (13). To this end, we write out (20) as two coupled difference equations as follows:

$$\begin{aligned} &AM(\hat{x}_1^{n+1/2} - \hat{x}_1^n) + \delta(C\hat{x}_1^{n+1/2} - \hat{x}_2^{n+1/2}) \\ &\quad = \delta AM\bar{L}_1\bar{z}^{n+1/2} \end{aligned} \tag{21}$$

$$\begin{aligned} &(AD - C)(\hat{x}_1^{n+1/2} - \hat{x}_1^n) + (\hat{x}_2^{n+1/2} - \hat{x}_2^n) + \delta AK\hat{x}_1^{n+1/2} \\ &\quad = \delta(AD - C)\bar{L}_1\bar{z}^{n+1/2} + \delta\bar{L}_2\bar{z}^{n+1/2} + \delta ABu^{n+1/2} \end{aligned} \tag{22}$$

Multiplying (22) by δ and adding the resulting equation to (21) yields

$$\begin{aligned} &A(M + \delta D + \delta^2 K)\hat{x}_1^{n+1/2} = (AM + \delta(AD - C))\hat{x}_1^n + \delta\hat{x}_2^n \\ &+\{\delta AM\bar{L}_1 + \delta^2(AD - C)\bar{L}_1 + \delta^2\bar{L}_2\}\bar{z}^{n+1/2} + \delta^2 ABu^{n+1/2} \end{aligned} \tag{23}$$

Of several possible choices for matrices A and B, we will examine the following two specific cases:

$$\begin{cases} a) \quad A = I\,, \quad C = D \\ b) \quad A = M^{-1}\,, \quad C = 0 \end{cases} \tag{24}$$

where the mass matrix M is nonsingular due to its physically positive definite nature since the kinetic energy of structural system is positive for any admissable motion. It is noted that the above two choices, although mathematically equivalent, lead to different computational implementations as discussed below.

The choice of (24a) reduces (23) to:

$$\begin{aligned} (M+\delta D+\delta^2 K)\hat{x}_1^{n+1/2} &= M\hat{x}_1^n + \delta\hat{x}_2^n + \delta^2 B u^{n+1/2} \\ &+\delta\{M\bar{L}_1 + \delta\bar{L}_2\}\bar{z}^{n+1/2} \end{aligned} \tag{25}$$

so that once $\hat{x}_1^{n+1/2}$ is computed, $\hat{x}_2^{n+1/2}$ is obtained from (22) rewritten as

$$\hat{x}_2^{n+1/2} = \hat{x}_2^n + \delta\hat{g}^n - \delta K\hat{x}_1^{n+1/2} \tag{26}$$

where

$$\hat{g}^n = Bu^{n+1/2} + \bar{L}_2\bar{z}^{n+1/2} \tag{27}$$

which is already computed in order to construct the right-hand side of (25). Hence, $K\hat{x}_1^{n+1/2}$ is the only additional computation needed to obtain $\hat{x}_2^{n+1/2}$. It is noted that neither any numerical differentiation nor matrix inversion is required in computing $\hat{x}_2^{n+1/2}$. This has been achieved through the introduction of the general transformation (13) and the particular choice of the parameter matrices given by (24a).

On the other hand, if one chooses the conventional representation (24b), the solution of $\hat{x}_1^{n+1/2}$ is obtained from (23)

$$\begin{aligned} (M+\delta D+\delta^2 K)\hat{x}_1^{n+1/2} &= (M+\delta D)\hat{x}_1^n + \delta M\hat{x}_2^n \\ +\delta\{(M+\delta D)\bar{L}_1 &+ \delta M\bar{L}_2\}\bar{z}^{n+1/2} + \delta^2 Bu^{n+1/2} \end{aligned} \tag{28}$$

Once $\hat{x}_1^{n+1/2}$ is obtained, $\hat{x}_2^{n+1/2}$ can be computed either by

$$\hat{x}_2^{n+1/2} = (\hat{x}_1^{n+1/2} - \hat{x}_1^n)/\delta - \bar{L}_1\bar{z}^{n+1/2} \tag{29}$$

which is not accurate due to the numerical differentiation to obtain $\dot{\hat{x}}_1^{n+1/2}$, or by (22)

$$\begin{aligned} \hat{x}_2^{n+1/2} &= \hat{x}_2^n + \delta\hat{g}^n - \delta M^{-1}K\hat{x}_1^{n+1/2} - \\ M^{-1}D(\hat{x}_1^{n+1/2} &- \hat{x}_1^n) + \delta M^{-1}D\bar{L}_1\bar{z}^{n+1/2} \end{aligned} \tag{30}$$

which involves two additional matrix-vector multiplications, when $D \neq 0$, as compared with the choice of $A = I$ and $C = D$. Thus (24a) is the preferred representation in a first-order form of the second-order structural dynamics equations (1) and is used in the remainder of this work.

Decoupling Of Difference Equations

We have seen in the previous section, instead of solving the first-order Kalman filtering equations of $2n$ variables for the structural dynamics systems (1), the solution of the implicit time-discrete estimator equation (25) of n variables can potentially offer a substantial computational saving by exploiting the reduced size

and sparsity of M, D and K. This assumes that $\bar{z}^{n+1/2}$ and $u^{n+1/2}$ are available, which is not the case since at the n^{th} time step

$$u^{n+1/2} = -\bar{Z}_1 \hat{x}_1^{n+1/2} - \bar{Z}_2 \hat{x}_2^{n+1/2} \tag{31}$$

$$\bar{z}^{n+1/2} = z^{n+1/2} - \bar{H}_1 \hat{x}_1^{n+1/2} - \bar{H}_2 \hat{x}_2^{n+1/2} \tag{32}$$

requires both $\hat{x}_1^{n+1/2}$ and $\hat{x}_2^{n+1/2}$ even if $z^{n+1/2}$ is assumed to be known from measurements or by solution of (1). Note in (32), the control gain matrices are transformed by

$$\bar{Z}_1 = Z_1 - Z_2 M^{-1} A^{-1} C, \qquad \bar{Z}_2 = Z_2 M^{-1} A^{-1}$$

There are two distinct approaches to decouple (25) and (26) as described in the following sections.

Discrete Time Update

For systems utilizing discrete-time (i.e. sample and hold) control, equations (31) and (32) become

$$u^{n+1/2} \simeq -\bar{Z}_1 \hat{x}_1^n - \bar{Z}_2 \hat{x}_2^n \tag{33}$$

$$\bar{z}^{n+1/2} \simeq z^n - \bar{H}_1 \hat{x}_1^n - \bar{H}_2 \hat{x}_2^n \tag{34}$$

The time integration step size of the estimator must then be equal to the sample rate of the control, while the continuous structural equations may also be integrated at the same rate or at some fraction of the sampling rate for simulation accuracy considerations. For the present purposes, we have assumed that the sampling interval is the same as the integration time stepsize.

Discrete time simulation is quite simple to implement as the control force and state corrections are treated with no approximation on the right-hand-side (RHS) of (25) and (26). Should continuous time simulation be required, a different approach is necessary.

Continuous Time Update

To simulate the system given in (25) and (26) in continuous time, strictly speaking, one must rearrange (25) and (26) so that the terms involving $\hat{x}_1^{n+1/2}$ and $\hat{x}_2^{n+1/2}$ are augmented to the left-hand-side (LHS) of the equations. However, this augmentation into the solution matrix $(M + \delta D + \delta^2 K)$ would destroy the computational advantages of the matrix sparsity and symmetry. Thus, a partitioned solution procedure has been developed for continuous time simulation as described in Park and Belvin[17]. The procedure, briefly outlined herein, maintains the control force and state correction on the RHS of the equations as follows.

First, $\hat{x}_1^{n+1/2}$ and $\hat{x}_2^{n+1/2}$ are predicted by

$$\hat{x}_{1p}^{n+1/2} = \hat{x}_1^n, \qquad \hat{x}_{2p}^{n+1/2} = \hat{x}_2^n \tag{35}$$

However, instead of direct substitution of the above predicted quantity to obtain $u_p^{n+1/2}$ and $\bar{z}_p^{n+1/2}$ based on (31) and (32), equation augmentations are introduced to improve the accuracy of $u_p^{n+1/2}$ and $\bar{z}_p^{n+1/2}$. Of several augmentation procedures that are applicable to construct discrete filters for the computations of $u^{n+1/2}$ and $\bar{z}^{n+1/2}$, we substitute (26) into (31) and (32) to obtain

$$\left\{ \begin{aligned} u^{n+1/2} &= -\bar{Z}_1 \hat{x}_1^{n+1/2} - \bar{Z}_2(\hat{x}_2^n - \delta K \hat{x}_1^{n+1/2} + \\ &\quad \delta B u^{n+1/2} + \delta \bar{L}_2 \bar{z}^{n+1/2}) \\ \bar{z}^{n+1/2} &= z^{n+1/2} - \bar{H}_1 \hat{x}_1^{n+1/2} - \\ &\quad \bar{H}_2(\hat{x}_2^n - \delta K \hat{x}_1^{n+1/2} + \delta B u^{n+1/2} + \delta \bar{L}_2 \bar{z}^{n+1/2}) \end{aligned} \right. \tag{36}$$

Rearranging the above coupled equations, one obtains

$$\begin{bmatrix} (I+\delta\bar{Z}_2 B) & \delta\bar{Z}_2\bar{L}_2 \\ \delta\bar{H}_2 B & (I+\delta\bar{H}_2\bar{L}_2) \end{bmatrix} \left\{ \begin{matrix} u^{n+1/2} \\ \bar{z}^{n+1/2} \end{matrix} \right\} =$$

$$\left\{ \begin{matrix} -\bar{Z}_2\hat{x}_2^n - (\bar{Z}_1 - \delta\bar{Z}_2 K)\hat{x}_1^{n+1/2} \\ z^{n+1/2} - \bar{H}_2\hat{x}_2^n - (\bar{H}_1 - \delta\bar{H}_2 K)\hat{x}_1^{n+1/2} \end{matrix} \right\} \tag{37}$$

which corresponds to a first order filter to reduce the errors in computing $\hat{x}_2 = M\dot{\hat{x}} + D\hat{x}$. A second-order discrete filter for computing u and $\bar{z}$ can be obtained by differentiating u and $\bar{z}$ to obtain

$$\left\{ \begin{array}{l} \dot{u} = -\bar{Z}_1\dot{\hat{x}}_1 - \bar{Z}_2\dot{\hat{x}}_2 \\ \dot{\bar{z}} = \dot{z} - \bar{H}_1\dot{\hat{x}}_1 - \bar{H}_2\dot{\hat{x}}_2 \end{array} \right. \tag{38}$$

and then substituting $\dot{\hat{x}}_1$ and $\dot{\hat{x}}_2$ from (17). Subsequently, (19) is applied to integrate the equations for u and $\bar{z}$ which yields

$$\begin{bmatrix} I+\delta\bar{Z}_2 B + \delta^2\bar{Z}_1 M^{-1}B & \delta(\bar{Z}_2\bar{L}_2 + \bar{Z}_1\bar{L}_1 + \delta\bar{Z}_1 M^{-1}\bar{L}_2) \\ \delta(\bar{H}_2 B + \delta\bar{H}_1 M^{-1}B) & I+\delta\bar{H}_1(\bar{L}_1 + \delta M^{-1}\bar{L}_2) + \delta\bar{H}_2\bar{L}_2 \end{bmatrix} \left\{ \begin{matrix} u^{n+1/2} \\ \bar{z}^{n+1/2} \end{matrix} \right\} =$$

$$\left\{ \begin{matrix} u^n \\ \bar{z}^n \end{matrix} \right\} - \delta \left\{ \begin{matrix} \bar{Z}_1 M^{-1}(\hat{x}_2^n - \delta K\hat{x}_1^{n+1/2} - D\hat{x}_1^{n+1/2}) + \bar{Z}_2 K\hat{x}_1^{n+1/2} \\ \bar{H}_1 M^{-1}(\hat{x}_2^n - \delta K\hat{x}_1^{n+1/2} D\hat{x}_1^{n+1/2}) + \bar{H}_2 K\hat{x}_1^{n+1/2} \end{matrix} \right\} + \left\{ \begin{matrix} 0 \\ z^{n+1/2} - z^n \end{matrix} \right\} \tag{39}$$

The net effects of this augmentation are to filter out the errors committed in estimating both $\hat{x}_1$ and $\hat{x}_2$. Solution of (39) for $u^{n+1/2}$ and $\bar{z}^{n+1/2}$ permits (25) and (26) to be solved in continuous time for $\hat{x}_1^{n+1/2}$ and $\hat{x}_2^{n+1/2}$. Subsequently, (19b) is used for $\hat{x}_1^{n+1}$ and $\hat{x}_2^{n+1}$.

The preceding augmentation (39) leads to an accurate estimate of the control force and state estimation error correction at the (n+1/2) time step. Although (39) involves the solution of an additional algebraic equation, the equation size is relatively small (size = number of actuators (m) plus the number of measurements (r)). Thus, (39) is an efficient method for continuous time simulation of the Kalman filter equations provided the size of (39) is significantly lower than the first order form of (4). The next section discusses the relative efficiency of the present method and the conventional first order solution. More details on the equation augmentation procedure (39) may be found in Park and Belvin[17].

Finally, it is noted that by following a similar time discretization procedure adopted for computing $\hat{x}_1^{n+1/2}$ and $\hat{x}_2^{n+1/2}$, the structural dynamics equation (1) can be solved by

$$\left\{ \begin{array}{l} (M+\delta D+\delta^2 K)x_1^{n+1/2} = Mx_1^n + \delta x_2^n + \delta^2 Bu^{n+1/2} \\ x_2^{n+1/2} = x_2^n + \delta Bu^{n+1/2} - \delta Kx_1^{n+1/2} \end{array} \right. \tag{40}$$

Thus, numerical solutions of the structural dynamics equation (1) and the filter equation (20) can be carried out within the second-order solution context, thus realizing substantial computational simplicity compared with the solution of first-order systems of equations (4) and the corresponding first-order estimator equations (6).

It is emphasized that the solutions of both the structural displacement x and the reconstructed displacement $\hat{x}$ employ the same solution matrix, $(M + \delta D + \delta^2 K)$. The computational stability of the present procedure can be examined as investigated in Park[18] and Park and Felippa[19,20]. The result, when applied to the present case, can be stated as

$$\delta^2\lambda_{\max} \leq 1 \tag{41}$$

where $\lambda_{\max}$ is the maximum eigenvalue of

$$(\lambda^2 I + \lambda \bar{Z}_2 B + \bar{Z}_1 M^{-1} B)y = 0 \tag{42}$$

Typically the control laws are formulated in terms of low-frequency response components, viz.,

$$B \propto G^T K G \tag{43}$$

for the displacement feedback case where G is a projection matrix that extracts only low-frequency components from the structural stiffness matrix. Hence, $\lambda_{\max}$ is in practice several orders of magnitude smaller than $\mu_{\max}$ of the structural dynamics eigenvalue problem:

$$\mu M y = K y \tag{44}$$

Considering that a typical explicit algorithm has its stability limit $\mu_{\max} \cdot h \leq 2$, the maximum step size allowed by (42) is in fact several orders of magnitude larger than allowed by any explicit algorithm.

Computational Efficiency

Solution of the Kalman filtering equations in second-order form is prompted by the potential gain in computational efficiency due to the beneficial nature of matrix sparsity and symmetry in the solution matrix of the second-order estimator equations. There is an overhead to be paid for the present second-order procedure, that is, the additional computations introduced to minimize the control force and state estimation error terms on the right-hand-side of the resulting discrete equations. The following paragraphs show the second-order solution is most advantageous for estimator models with sparse coefficient matrices M, D and K.

Solution of the first order Kalman filter equation (6) or the second-order form (25-26, 39) may be performed using a time discretization as given by (19). For linear time invariant (LTI) systems, the solution matrix is decomposed once and subsequently upper and lower triangular system solutions are performed to compute the estimator state at each time step. Thus, the computations required at each time step result from calculation of the RHS and subsequent triangular system solutions. For the results that follow, the number of floating point operations are estimated for LTI systems of order O(N). In addition, it is assumed that the mass, damping and stiffness matrices (M, D and K) are symmetric and banded with bandwidth αN, where $0 \leq \alpha \leq (0.5 - \frac{1}{2N})$.

The first-order Kalman filter equation (6) requires $(4N^2 + 2Nr + O(N))$ operations at each time step. The discrete time second-order Kalman filter solution (25-26, 33-34) require $(8\alpha^2 N^2 + 2\alpha N^2 + 3Nm + 4Nr + O(N))$ operations and the continuous time second-order Kalman filter (25-26, 39) require $(8\alpha N^2 + 2\alpha N^2 + 5Nm + 6Nr + (r+m)^2 + O(N))$ operations at each time step. To examine the relative efficiency of the first-order and second-order forms, several cases are presented as follows.

First, a worst case condition is examined whereby M, D and K are fully populated ($\alpha = 0.5 - \frac{1}{2N}$) and $r = m = N$. Only for this extreme condition with large numbers of sensors and actuators relative to the system order, the first order Kalman filter becomes somewhat more efficient than the second-order discrete Kalman filter solution presented herein.

For typical structural systems, M and K are almost always banded. In addition, the number of sensors and actuators is usually small compared to the system order N. If the number of actuators (m) and the

number of measurements (r) are proportional to the bandwidth ($r = m = \alpha N$), the second-order discrete Kalman filtering equations become computational attractive as long as $\alpha \leq 0.394$. It should be noted that the larger the size of the structural systems, the smaller the bandwidth becomes, with the range of α to be $0.05 \leq \alpha \leq 0.15$.

Finally, for the special case of modal-based structural models, one has $\alpha \rightarrow 0$. For this case, as long as sensors and actuators are sufficiently smaller than the modal degrees of freedom, the present second-order state estimator can be substantially more efficient than the classical first-order form. This is because the conventional state space-based estimator must deal with a fully coupled nonsymmetric $2N \times 2N$ system whereas the present second-order estimator deals with a diagonal $N \times N$ system. A more detailed discussion can be found in Belvin[9].

Implementation and Numerical Evaluations

The second-order discrete Kalman filtering equation derived in (25) and (26) have been implemented along with the stabilized form of the controller $\mathbf{u}$ and the filtered measurements $\bar{\mathbf{z}}$ in such a way the estimator computational module can be interfaced with the partitioned control-structure interaction simulation package developed previously by Belvin9, Park and Belvin[17] Alvin and Park[21]. It is emphasized that the solution procedure of the present second-order discrete Kalman filtering equations (25) and (26) follows exactly the same steps as required in the solution of symmetric, sparse structural systems. It is this attribute that makes the present discrete filter attractive from the simulation viewpoint. For a succinct comparison between the present CSI simulation procedure and conventional state space-based simulation procedures, the equations that need to be implemented in both of the procedures are summarized below.

Partitioned Control-Structure Interaction Equations

The partitioned procedure for simulating the control-structure interaction problems developed in Belvin[9] and Park and Belvin[17] exploits the second-order diferential equation form whenever possible as shown below.

$$\left\{\begin{array}{lll}
\text{Structure:} & a) & \mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \\
 & & \quad \mathbf{q}(0) = \mathbf{q}_0, \qquad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \\
\text{Sensor Output:} & b) & \mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \\
\text{Estimator:} & c) & \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \left\{ \begin{matrix} \dot{\tilde{\mathbf{q}}} \\ \dot{\tilde{\mathbf{p}}} \end{matrix} \right\} + \begin{bmatrix} \mathbf{D} & -\mathbf{I} \\ \mathbf{K} & 0 \end{bmatrix} \left\{ \begin{matrix} \tilde{\mathbf{q}} \\ \tilde{\mathbf{q}} \end{matrix} \right\} = \left\{ \begin{matrix} 0 \\ \mathbf{f} + \mathbf{B}\mathbf{u} \end{matrix} \right\} + \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{bmatrix} \gamma \\
 & & \quad \tilde{\mathbf{q}}(0) = 0, \quad \tilde{\mathbf{p}}(0) = \mathbf{f}(0) + \mathbf{B}\mathbf{u}(0) \\
\text{Control Force:} & d) & \dot{\mathbf{u}} + \mathbf{F}_2\mathbf{M}^{-1}\mathbf{B}\mathbf{u} = \mathbf{F}_2(\mathbf{M}^{-1}\dot{\tilde{\mathbf{p}}} + \mathbf{L}_2\gamma) + \mathbf{F}_1\dot{\tilde{\mathbf{q}}} \\
\text{Estimation Error:} & e) & \dot{\gamma} + \mathbf{H}_v\mathbf{L}_2\gamma = \dot{\mathbf{z}} - \mathbf{H}_v\mathbf{M}^{-1}(\dot{\tilde{\mathbf{p}}} - \mathbf{B}\mathbf{u}) - \mathbf{H}_d\dot{\tilde{\mathbf{q}}}
\end{array}\right. \qquad (45)$$

In addition, notice that the control laws ($\mathbf{u}$) and the estimation error (γ) are parabolically stabilized and solved in a separate software module from the estimator and the structural analyzers, thus effectively rendering a computaionally efficient and accurate procedure.

Conventional Control-Structure Interaction Equatioons

In contrast to the partitioned procedure summarized above, conventional control-structure interaction simulation employs a first-order differential equation form as shown below, thus requiring the solution of $2n \times 2n$)-system equations for structures and the observer. In addition, the control laws and the estimation errror are not stabilized, which can give rise to an accumulation of computational errors.

$$\left\{\begin{array}{lll} \text{Structure:} & a) & \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{f} + \bar{\mathbf{B}}\mathbf{u} + \bar{\mathbf{G}}\mathbf{w} \\ & & \mathbf{x}(0) = \mathbf{x}_0 \\ \text{Sensor Output:} & b) & \mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \\ \text{Estimator:} & c) & \dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{E}\mathbf{f} + \bar{\mathbf{B}}\mathbf{u} + \mathbf{L}\gamma \\ & & \tilde{\mathbf{x}}(0) = 0 \\ \text{Control Force:} & d) & \mathbf{u} = -\mathbf{F}\tilde{\mathbf{x}} \\ \text{Estimation Error:} & e) & \gamma = \mathbf{z} - \mathbf{H}\tilde{\mathbf{x}} \end{array}\right. \tag{46}$$

where

$$\mathbf{x} = \left\{ \begin{array}{c} \mathbf{q} \\ \dot{\mathbf{q}} \end{array} \right\}, \quad \tilde{\mathbf{x}} = \left\{ \begin{array}{c} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{array} \right\}$$

and

$$\mathbf{H} = [\mathbf{H}_d \quad \mathbf{H}_v], \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \\ \mathbf{M}^{-1} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} 0 \\ \mathbf{M}^{-1}\mathbf{B} \end{bmatrix}, \quad \mathbf{F} = [\mathbf{F}_1 \quad \mathbf{F}_2]$$

Numerical Experiments

The first example is a truss beam shown in Fig. 1, consisting of 8 bays with nodes 1 and 2 fixed for cantilevered motions. Actuator and sensor locations, as well as their orientation, are given in Table 1. In the numerical experiments reported herein, we have relied on the Matlab software package[22] for the synthesis of both the control law gains and the discrete Kalman filter gain matrices. Figures 2, 3 and 4 show the vertical displacement time response at node 9 for open-loop, full state feedback, and dynamically compensated feedback cases, respectively. In the present pepper, a full state feedback corresponds to the case for which the number of sensors are the same as the total system degrees of freedom whereas the dynamically compensated case corresopnds to a smaller number of sensors as compared with the total system degrees of freedom. Note the effectiveness of the dynamically compensated feedback case with four actuators and six sensors as indicated in Table 1 by the present second-order discrete Kalman filtering equations as compared with the full state feedback cases.

Figure 5 illustrates a testbed model of an Earth-pointing satellite. For vibration control, 18 actuators and 18 sensors are configured throughout the system; their locations are provided in Tables 2 and 3. Figures 6, 7, and 8 are a representative of the responses for open-loop, full state feedback, and dynamically compensated cases, respectively. In both examples, the estimator states are the estimated physical displacements and generalized momenta as previously developed, and thus the number of effective states is equal to $2N$, where N is the number of physical displacement variables of the second-order structural system. Therefore, the Kalman filter for the truss example has 108 states, and the filter for the satellite has 1164 states, a substantial increase over typical estimator orders for such systems. Further simulations with the present procedure should

shed light on the performance of dynamically compensated feedback systems for large-scale systems as they are computationally more feasible than heretofore possible.

The computational overhead associated with the full state feedback vs. the use of a dynamic compensation scheme by the present Kalman filtering equations is reported in Table 4. It is seen that the use of the present second-order discrete Kalman filtering equations for constructing dynamically compensated control laws adds computational overhead, only an equivalent of open-loop transient analysis of symmetric sparse systems of order N instead of $2N \times 2N$ dense systems. This is evidenced in Table 4 in that the normalized CPU time for the dynamically compensated case (designated as K. Filter) is 63.16 whereas the total CPUs for the full state feedback case (FSFB) plus that of the open loop dynamic response (Transient) is 64.18.

Summary

The present paper has addressed the advantageous features of employing the same direct time integration algorithm for solving the structural dynamics equations and for integrating the associated continuous Kalman filtering equations. The time discretization of the resulting Kalman filtering equations is further facilitated by employing a canonical first-order form via a generalized momenta. When used in conjunction with the previously developed stabilized form of control laws in Park and Belvin[17], the present procedure offers a substantial computational advantage over the simulation methods based on a first-order form when computing with large (i.e. nearly full system dynamics) and sparse estimator models.

In order to minimize the deleterious effect of numerical damping and phase distortion in the solution of the discrete Kalman filtering equations, the trapezoidal rule is employed. This is due to the wellknown fact that the trapezoidal rule conserves the system energy with minimum phase error among all the time integration formulas of second-order accuracy[15,16]

Computational stability of the present solution method for the filter equation has been assessed based on the stability analysis result of partitioned solution procedures[18]. To obtain a sharper estimate of the stable integration step size, a more rigorous computational stability analysis is being carried out and will be reported in the future.

Acknowledgements

The work reported herein was supported by a grant from NASA/Johnson Space Center, NAG 9-574 and a grant from NASA/Langley Research Center, NAG1-1021. The authors thank Dr. John Sunkel of NASA/Johnson Space Center and Dr. Jer-Nan Juang of NASA/Langley Research Center who encouraged us to work on second-order observers.

References

1. Kwakernaak, H. and Sivan, R., *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.
2. Hughes, P. C. and Skelton, R. E., "Controllability and observability of linear matrix second-order systems," *Journal of Applied Mechanics, Trans. ASME*, Vol. 47, 1980, 415-420.
3. Arnold, W. F. and Laub, A. J., "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," *Proceedings of the IEEE*, Vol. 72, No. 12, 1984, 1746-1754.
4. Bender, D. J. and Laub, A. J., "Controllability and Observability at Infinity of Multivariable Linear Second-Order Models," *IEEE Transactions on Automatic Control*, Vol. AC-30, 1985, 1234-1237.

5. Oshman, Y., Inman, D. J. and Laub, A. J., "Square-Root State Estimation for Second-Order Large Space Structures Models," *Journal of Guidance, Control and Dynamics*, Vol. 12, no. 5, 1989, 698-708.

6. Belvin, W. K. and Park, K. C., "On the State Estimation of Structures with Second Order Observers," *Proc. the 30th Structures, Dynamics and Materials Conference*, AIAA Paper No. 89-1241, 1989.

7. Belvin, W. K. and Park, K. C., "Structural Tailoring and Feedback Control Synthesis: An Interdisciplinary Approach," *Journal of Guidance, Control and Dynamics*, Vol. 13, No. 3, 1990, 424-429.

8. Juang, J. N. and Maghami, P. G., "Robust Eigensystem Assignment for Second-Order Estimators," *Proc. of the Guidance, Navigation and Control Conference*, AIAA Paper No. 90-3474, 1990.

9. Belvin, W. K., "Simulation and Interdisciplinary Design Methodology for Control-Structure Interaction Systems," PhD Thesis, Center for Space Structures and Controls, University of Colorado at Boulder, CO., Report No. CU-CSSC-89-10, July 1989.

10. Kalman, R. E., "On the General Theory of Control Systems," *Proc. 1st International Congress on Automatic Control*, Butterworth, London, Vol. 1, 1961, 481-491.

11. Kalman, R. E. and Bucy, R. S., "New results in linear filtering and prediction theory," *Trans. ASME J. Basic Engineering*, Vol. 83, 1961, 95-108.

12. Hashemipour, H. R. and Laub, A. J., "Kalman filtering for second-order models," *J. Guidance, Control and Dynamics*, Vol. 11, No. 2, 1988, 181-185.

13. Jensen, P. S., "Transient Analysis of Structures by Stiffly Stable Methods," *Computers and Structures*, Vol. 4, 1974, 67-94.

14. Felippa, C. A. and Park, K. C., "Computational Aspects of Time Integration Procedures in Structural Dynamics, Part 1: Implementation," *Journal of Applied Mechanics*, Vol. 45, 1978, 595-602.

15. Dahlquist, G., "A Special Stability Problem for Linear Multi-step Methods," *BIT*, 3, 1963, pp. 27-43.

16. Park, K. C., "Practical Aspects of Time integration," *Comp. & Struct.*, 7, 1977, 343-353.

17. Park, K. C. and Belvin, W. K., "Stability and Implementation of Partitioned CSI Solution Procedures," *Journal of Guidance, Control, and Dynamics*, **14**, January-February 1991, 59-67.

18. Park, K. C., "Partitioned Analysis Procedures for Coupled-Field Problems: Stability Analysis," *Journal of Applied Mechanics*, Vol. 47, 1980), 370-378.

19. Park, K. C. and Felippa, C. A., "Partitioned Analysis of Coupled Systems," in: *Computational Methods for Transient Analysis*, T. Belytschko and T. J. R. Hughes (eds.), Elsevier Pub. Co., 1983, 157-219.

20. Park, K. C. and Felippa, C. A.,"Recent Developments in Coupled-Field Analysis Methods," in: *Numerical Methods in Coupled Systems*, Lewis, R. W. et al(editors), John Wiley & Sons, 1984, 327-352.

21. K. F. Alvin and K. C. Park, "Implementation of A Partitioned Algorithm for Simulations of Large CSI Problems," Center for Space Structures and Controls, University of Colorado at Boulder, CO., Report No. CU-CSSC-91-4, March 1991.

22. *PRO-Matlab User's Guide*, The MathWorks, Inc., 1989.

Table 1a

Actuator Placement for Truss Example Problem

Actuator	Node	Component
1	2	y
2	18	y
3	9	y
4	9	x

Table 1b

Sensor Placement for Truss Example Problem

Sensor	Type	Node	Component
1	Rate	2	y
2	Rate	18	y
3	Rate	9	y
4	Rate	9	x
5	Position	9	y
6	Position	9	x

Table 2

Actuator Placement for EPS Example Problem

Actuator	Node	Component
1	97	x
2	97	z
3	96	x
4	96	z
5	65	y
6	68	y
7	59	y
8	62	y
9	45	y
10	45	z
11	70	y
12	70	z
13	95	x
14	95	y
15	95	z
16	95	ϕ_x
17	95	ϕ_y
18	95	ϕ_z

Table 3

Sensor Placement for EPS Example Problem

Sensor	Type	Node	Component
1	Rate	97	x
2	Rate	97	z
3	Rate	96	x
4	Rate	96	z
5	Rate	65	y
6	Rate	68	y
7	Rate	59	y
8	Rate	62	y
9	Rate	45	y
10	Rate	45	z
11	Rate	70	y
12	Rate	70	z
13	Position	95	x
14	Position	95	y
15	Position	95	z
16	Position	95	ϕ_x
17	Position	95	ϕ_y
18	Position	95	ϕ_z

Table 4

CPU Results for ACSIS Sequential and Parallel Versions

	Problem		
Model	Type	Sequential	Parallel
54 DOF	Transient	4.5	5.6
Truss	FSFB	9.4	10.2
	K. Filter	13.0	10.7
582 DOF	Transient	98.6	100.3
EPS7	FSFB	190.2	294.5
	K. Filter	284.2	321.5

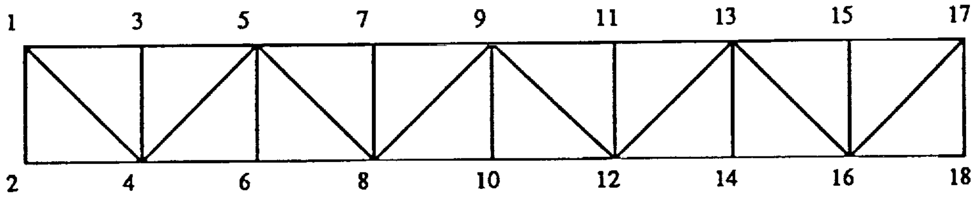


Figure 1: TRUSS BEAM PROBLEM

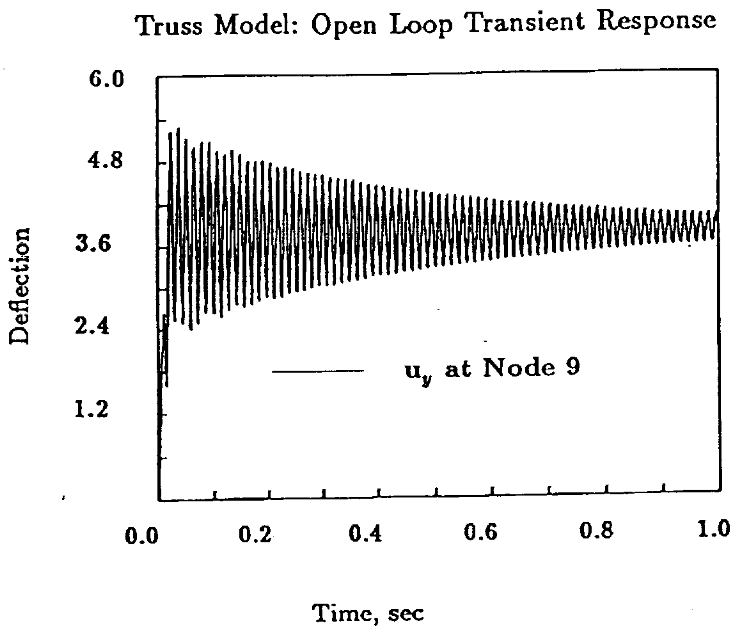


Figure 2: Truss Transient Response

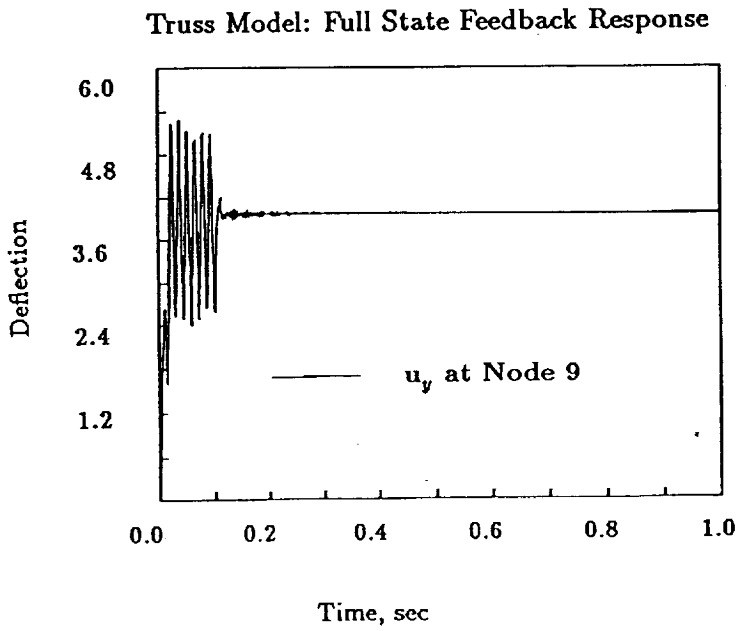


Figure 3: TRUSS FULL STATE FEEDBACK RESPONSE

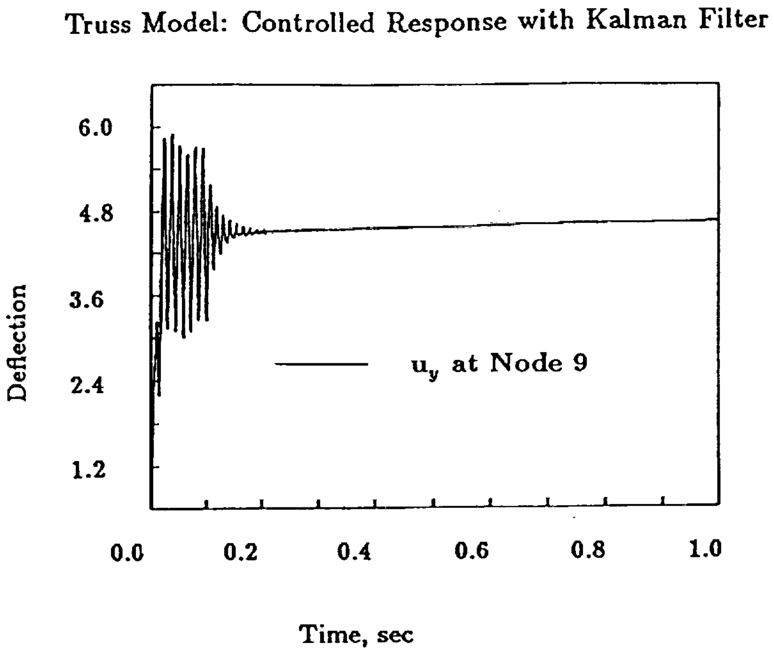


Figure 4: Truss Response with Filter

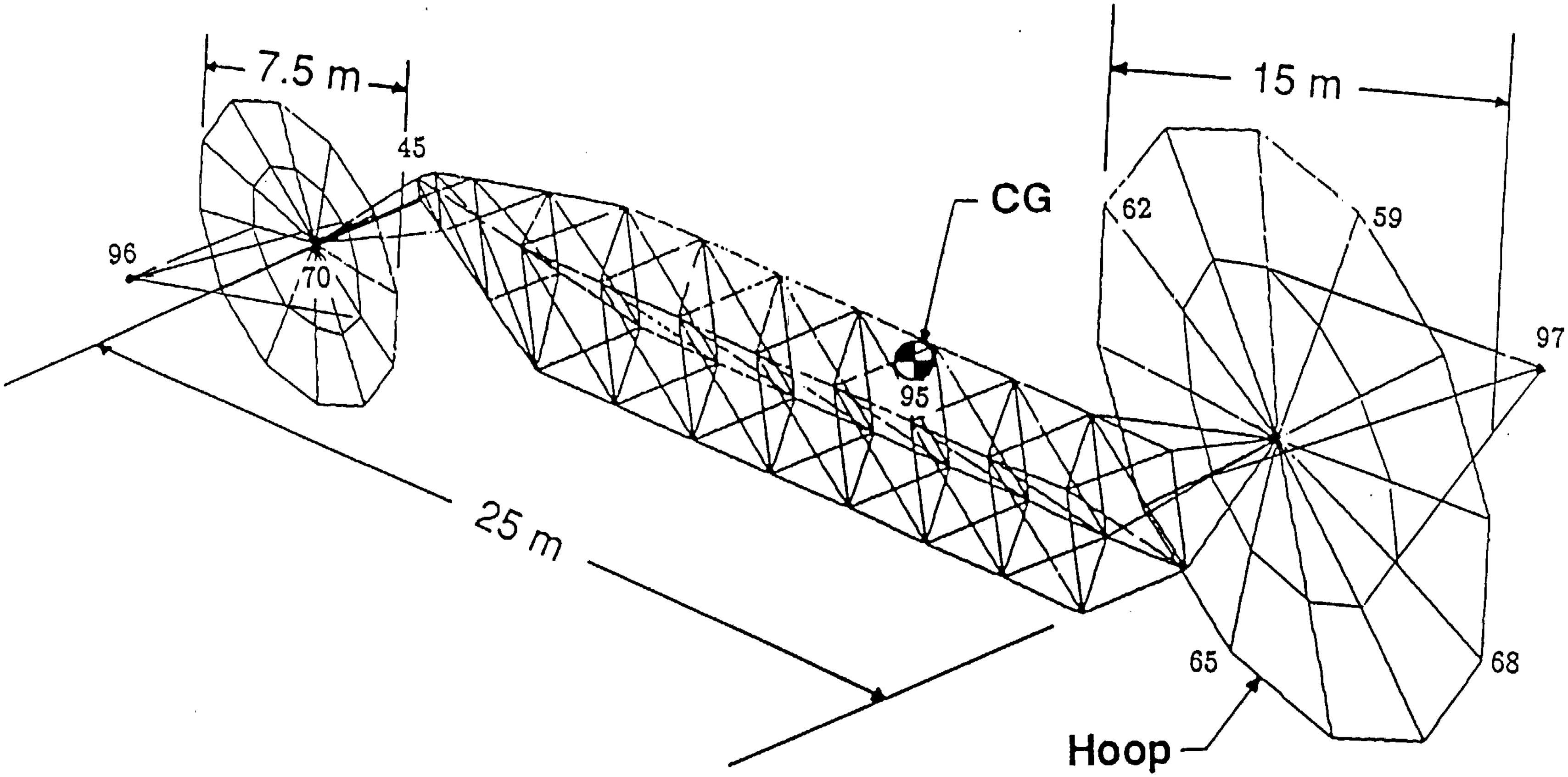


Figure 5: GENERIC EARTH POINTING SATELLITE EXAMPLE

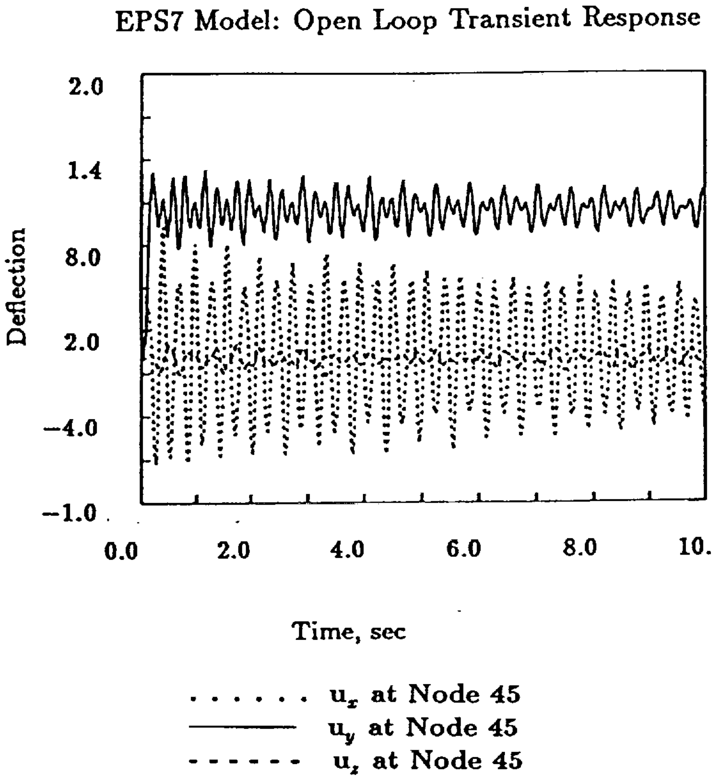


Figure 6: EPS TRANSIENT RESPONSE

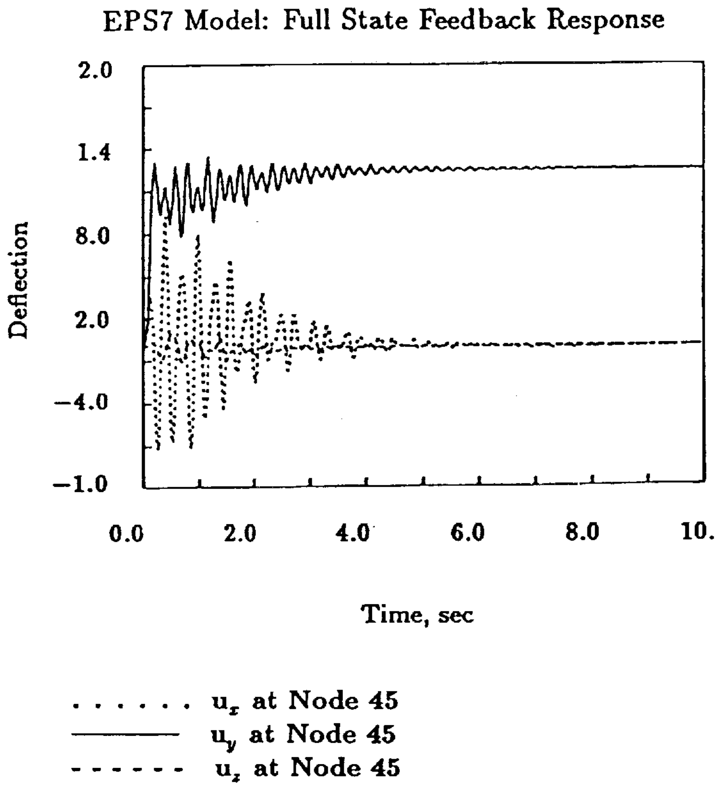


Figure 7: EPS FULL STATE FEEDBACK RESPONSE

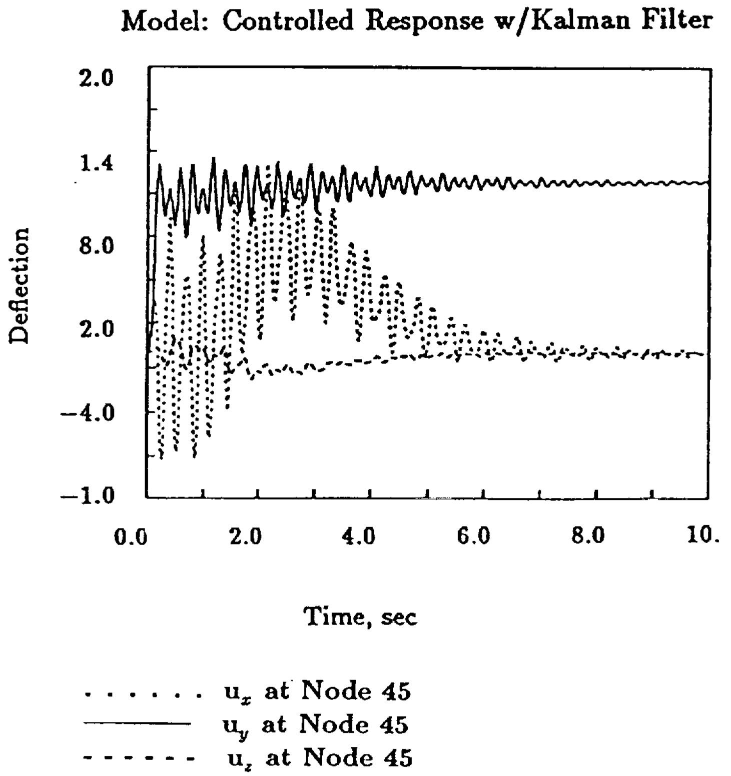


Figure 8: EPS RESPONSE with KALMAN FILTER

Dynamics of Adaptive Structures: Design through Simulations

K. C. Park and S. Alexander
Department of Aerospace Sciences Engineering and
Center for Space Structures and Controls
University of Colorado, Campus Box 429
Boulder, CO 80309-429

1. INTRODUCTION

Mechanisms and static stress analyses have long been the major considerations in the design of many articulated structures or adaptive structures in the past. However, high-performance requirements on these structures have added the dynamics considerations as a new added design criterion in recent years. This is especially true in the design of adaptive or deployable space structures that involve the combined phenomena of the orbital mechanics, structural configuration changes and flexible vibrations in a coupled manner. Hence, little attention has been given to, in the design of reconfigurable flexible space structures, the influence of the accompanying dynamics during the maneuvering as an integral part of the design requirements.

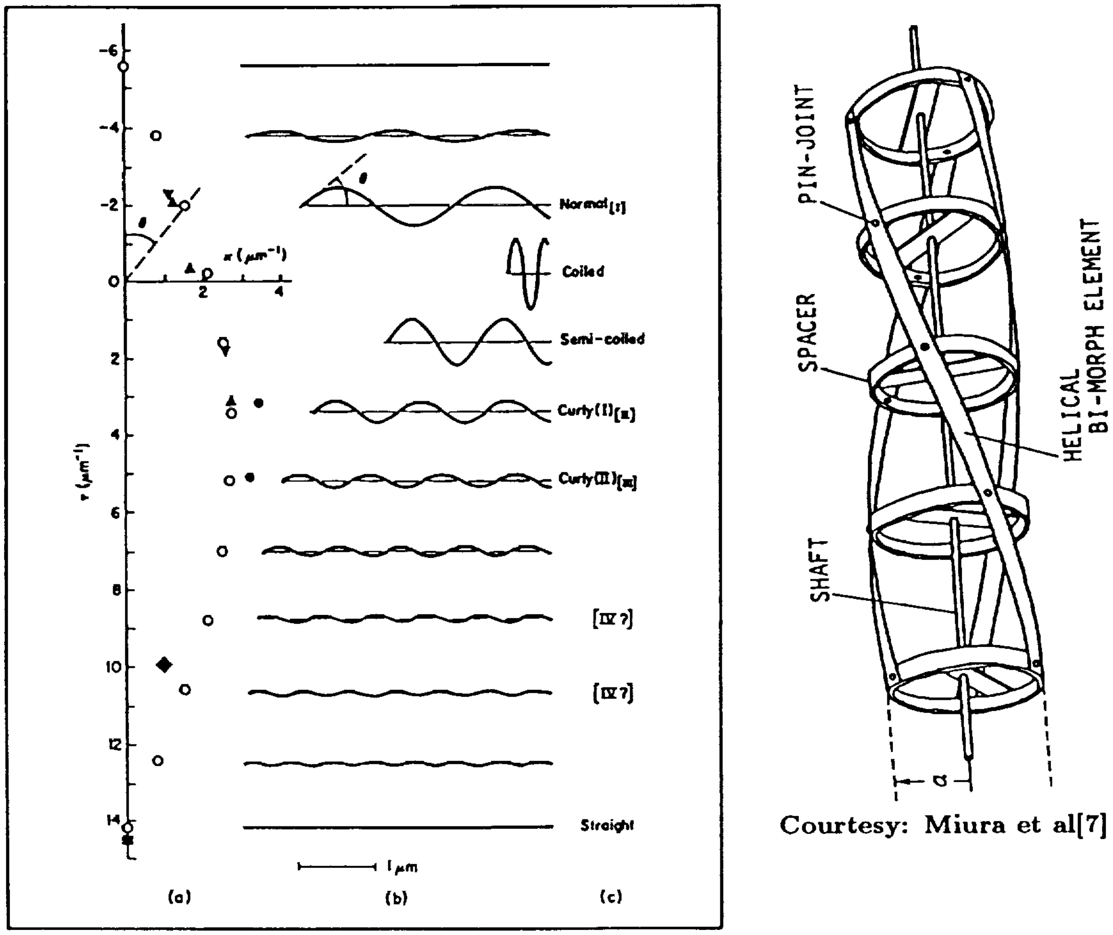
The adaptations of human bodies, animals and bacteria to spatial dynamical motions have been previously studied[1-3]. Recently, several investigators developed the so-called angular momentum preserving rotational maneuvering control algorithms and applied them to robotics and spacecraft attitude controls[4-6]. As a result, the intrinsic adaptations of the momentum conservation (violation for that matter) laws by spring board divers, ice skaters as well as gymnasts are well understood, which have been subsequently utilized for the design of space robotics maneuvering and space rendezvous scenarios. These studies have dealt mostly with rigid bodies linked by frictionless joints and focused on the development of various control algorithms for nonholonomic rigid dynamical system.

The use of a helical bi-morph actuator/sensor concept[7] by mimicking the change of helical waveform in bacterial flagella is perhaps the first application of bacterial motions (living species) to longitudinal deployment of space structures. However, no dynamical considerations were analyzed to explain the waveform change mechanisms[3, 7]. The objective of the present paper is to review various deployment concepts from the dynamics point of view and introduce the dynamical considerations from the outset as part of design considerations. Specifically, the impact of the incorporation of the combined static mechanisms and dynamic design considerations on the deployment performance during the reconfiguration stage is studied in terms of improved controllability, maneuvering duration and joint singularity index. It is shown that intermediate configurations during articulations play an important role for improved joint mechanisms design and overall structural deployability.

2. EXAMPLES OF ADAPTIVE STRUCTURES

2.1 Bacterial Flagella

In studying the chemotaxis of bacteria such as *Salmonella*, scientists discovered that their motions are intertwined with smooth swimming interrupted by short periods of tumbling[3]. In particular, the change in waveforms do not follow the intuitive way, vz., from one normal wave form to the adjacent discrete wave state. Instead, the transition of the waveform jump from one wave sometimes to its half-length wave. Calladine[3] conjectured that the intermittent existence of bi-stable subunits along the helical flagella structure are responsible for the formation of partly stable curly right-handed helices. It is these bi-stable subunits that cause jumps in the waveform formation.



Courtesy: Miura et al[7]

Courtesy: Caladine[3]

Fig. 1 Possible Waveforms of Flagella of Salmonella

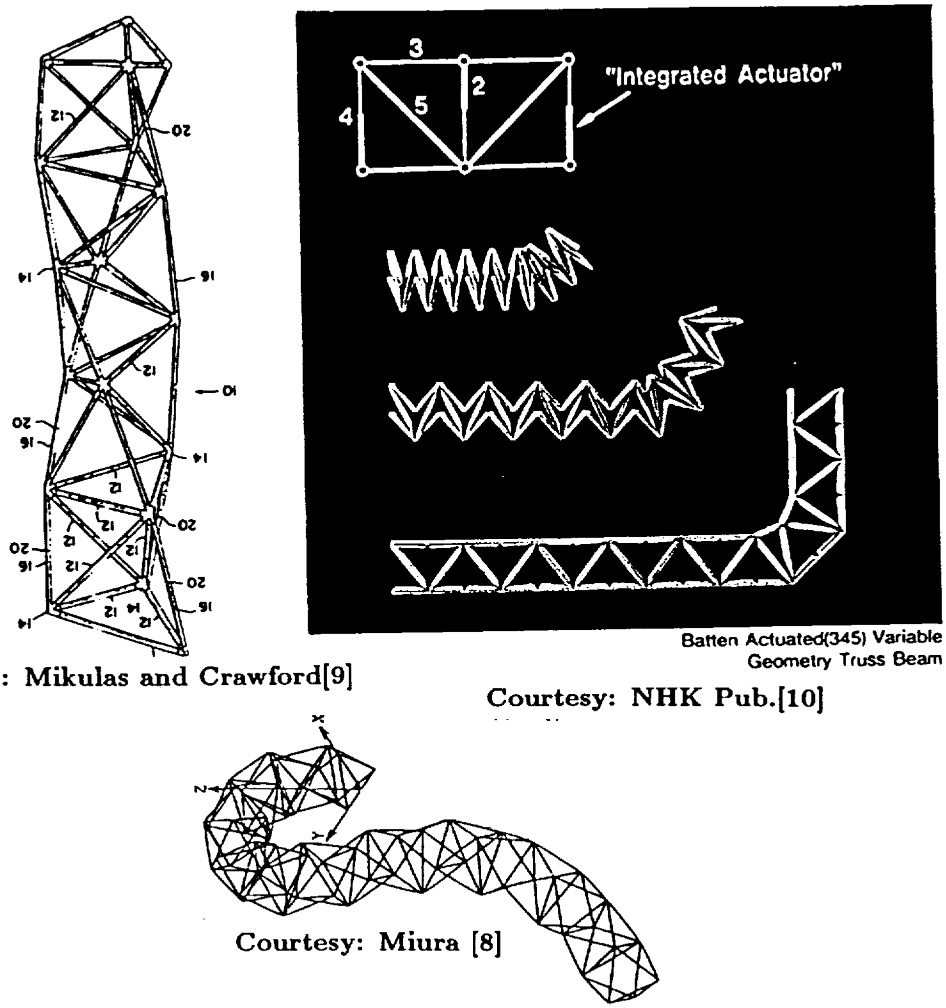
From the mechanical deployment perspectives, the large motions due to the jumps in waveform change in bacterial flagella pose the following questions: 1) how can such large motions be possible what are the sources of the torques that make such large

motions possible?; 2) are those motions created by minimizing the energy requirements or by triggering unstable motion paths so that the energy need remains minimal?; 3) can the large motion phenomenon be explained solely by quasi-static equilibrium considerations or be explained only by the dynamical considerations?

Experiments as well as analytical studies[3] so far identified twelve polymorphic helical forms with a tubular chains of 20 nanometer in diameter as shown in Fig. 1.

2.2 Reconfigurable Truss Beams

Figure 2 illustrates three representative reconfigurable truss beams. The sequentially deployable maneuvering tetrahedral beam is shown in Fig. 2a and can only be deployed sequentially, hence can't simulate the jumps in waveform of the bacterial flagella.



Courtesy: Mikulas and Crawford[9]

Batten Actuated(345) Variable Geometry Truss Beam

Courtesy: NHK Pub.[10]

Courtesy: Miura [8]

Fig. 2 Various Reconfigurable Truss Beams

By inserting actuator-encorder pairs into some of the truss members as shown in the variable geometry truss (Fig. 2b), it is possible to shape the beam as desired. The batten actuated beam as shown in Fig. 2c is perhaps the simplest reconfigurable truss. In both the last two cases, the actuators may be viewed as bi-stable subunits which, unlike for the case of tumbling motions in flagella case, do require control forces.

3. NONHOLONOMICALLY CONTROLLED RECONFIGURABLE STRUCTURES

The equations of motion for nonholonomically controlled reconfigurable structures can be written as

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{f}(t) - \mathbf{S}(\mathbf{q}) + \mathbf{B}\mathbf{u} + \mathbf{C}\lambda \\ \mathbf{p} &= \mathbf{M}\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q}) \end{aligned} \tag{1}$$

with the constraints:

$$\begin{aligned} \Phi_K(\mathbf{q}) &= 0 \quad \mathbf{C} = \frac{\partial \Phi_K}{\partial \mathbf{q}} \\ \Phi_N(\mathbf{q}) &= 0 \quad \mathbf{B} = \frac{\partial \Phi_K}{\partial \mathbf{q}} \end{aligned} \tag{2}$$

In the above equations, p is the generalized momenta, f is the applied force, S is the internal force, u and λ are the nonholonomic and kinematic contraint forces, M is the generalized inertia matrix, D is the damping operator, and $\Phi_K(\mathbf{q})$ and $\Phi_N(\mathbf{q})$ are system kinematic and nonholonomic constraint equations. It should be noted that both u and λ can be augmented with active control forces, when necessary.

Figure 3 illustrates a design example that involves the sizing of the double moemnt gyros[11] for effecting the maneuvering as well as the necessary vibration control. The moment gyros can in turn be made of from micro to mini sizes [12], depending upon the torque requirements. In this particular example, the task is to shape the articulated straight beam to form an hexagonal polygonal structure in space or can be shaped to form a helix if desired. Therefore, the role of gyros is to perform triple tasks concurrently: maneuvering, vibration control, and if necessary bi-stable units for easy articulation.

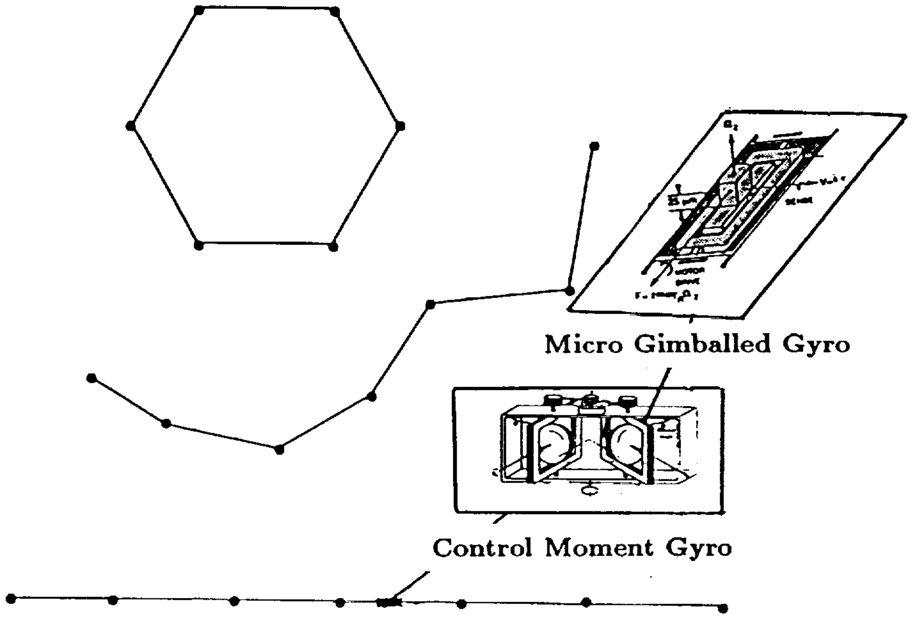


Fig. 3 Articulation of Beam-Like Structure via Control Moment Gyros

It should be mentioned that, for three rigidly linked planar maneuvering that conserves the angular momentum, the problem has been analyzed in [6]. It is for flexible cases the solution can be complicated. These and solutions of other related problems will be reported at the conference.

REFERENCES

1. Frolich, C., "Do Springboard Divers ViolateAngular Momentum Conservation?," American Journal of Physics, Vol. 47, 1979, 583-592.

2. Kane, T. R. and Scher, M. P., "A Dynamical Explanation of the Falling Cat Phenomenon," International Journal of Solids and Structures, Vol. 5, 1969, 663-670.

3. Calladine, C. R., "Change of Waveform in Bacterial Flagella: The Role of Mechanics at the Molecular Level," Journal of Molecuar Biology, Vol. 118, 1978, 457-479.

4. Spofford, J. R. and Akin, D. L., "Redundancy Control of a Free-Flying Telerobot," Journal of Guidance, Control and Dynamics, Vol. 13, may-June 1990, 515-523.

5. Misawa, M., Yasaka, T. and Miyake, S., "Analytical and Experimental Investigations for Satellite Antenna Deployment Mechanisms," Journal of Spacecraft and Rockets, Vol. 26, May-June 1989, 181-187.

6. Reyhanoglu, M. and McClamroth, N. H., "Reorientation of Space Multibody Systems Maintaining Zero Angular Momentum," Proc. AIAA Guidance, Navigation and Control Conference, 1991, Paper No. AIAA-91-2747-CP.

7. Miura, K., Sakamaki, M. and Ono, Y., "A New Concept of Actuator with Large Displacements," Proc. the Third International Conference on Adaptive Structures, San Diego, November 9-11, 1992.

8. Miura, Koryo, "Adaptive Structures Research at ISAS, 1884-1990," Journal of Intelligent Material Systems and Structures, Vol. 3, 1992, 54-74.

9. Mikulas, M. M. and Crawford, R. F., "Sequentially Deployable Maneuverable Tetrahedral Beam," U.S. Patent No. 4,557,097, December 10, 1985.

10. *The Space Age, Vol. 6: Return to the Moon*, NHK Publication Assn., Tokyo, Japan, 1993, p. 85.

11. Yang, L. F., Mikulas, M., Park, K. C. and Su, R., "Slewing Maneuvers and Vibration Controls of Space Structures by Combined Feedforward and Feedback Moment Gyro Controls," *Proc. 1993 AIAA SDM Conference*, Paper No. AIAA 93-1675, April 19-21 1993, LaJolla, CA.

12. Barbour, N. M., Elwell, J. M. and Setterlund, R. H., "Inertial Instruments: Where to Now?," *Proc. 1992 AIAA Guidance, Navigation and Control Conf.*, Paper No. AIAA-92-4414-CP, 566-574.